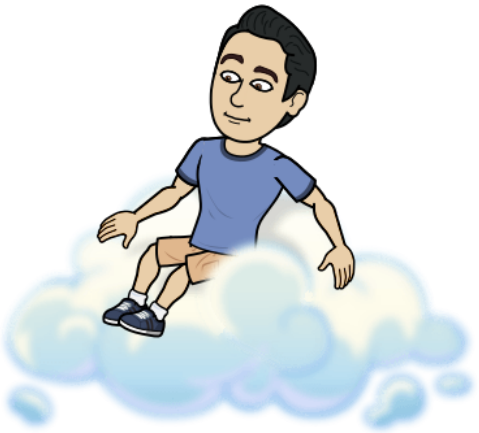
IES Villablanca

Comunidad de Madrid | CONSEJERÍA DE EDUCACIÓN, UNIVERSIDADES, CIENCIA Y PORTAVOCÍA

FONDO SOCIAL EUROPEO



Implantación completa de sistemas internos y de gestión de una empresa de alojamiento web

Con la inestimable colaboración de: clouding.io axarnet

Pablo González Troyano

Proyecto Fin de Ciclo - DocID:0

2º Administración de Sistemas Informáticos en Red

gonzaleztroyano.es/pfc

Foto de portada: Bitmoji

Autor y año de publicación: Pablo González Troyano, 2022

A todas esas maravillosas personas que me rodean,
sin su apoyo y comprensión este proyecto
no hubiera sido posible.

Índice

1. Introducción y justificación del proyecto

Se propone la creación de una empresa ficticia de *hosting* (alojamiento web) bajo el nombre comercial glez.cloud. “glez” viene dado por mi apellido, González; “cloud” viene dado por el enfoque comercial del servicio.

Los servicios gestionados de hosting están a la orden del día. Todas las empresas necesitan ese escaparate virtual que es la web. Estos sitios web necesitan un alojamiento. Ahí es donde entra GLEZ.CLOUD. GLEZ.CLOUD es un servicio de alojamiento premium, donde el cliente será atendido siempre por personas de verdad, nada de respuestas automáticas o tener que navegar por interminables artículos de ayuda.

El equipo de expertos de GLEZ.CLOUD estará siempre disponible para ayudar a los clientes en caso de necesitarlo. Cuando un cliente adquiera un plan de hosting (o cualquier otro de los servicios ofrecidos) un miembro del equipo se encargará de configurar para este la web y la base de datos, si la necesitara, así como aconsejarte si tuviera alguna duda. En GLEZ.CLOUD los *Success Strategist*[1] son los encargados de ayudar en todo lo que necesiten a los clientes. Tienen en *background* técnico necesario para proponer soluciones efectivas a los clientes, así como una completa comprensión de la casuística estratégica y comercial de los clientes.

Este hosting está orientado a empresas (mayormente PYMES) que necesiten un hosting confiable, y no dispongan de un equipo IT.

[1] Término muy utilizado por empresas de servicios para definir el personal que se dedica a asegurar el éxito de los clientes en el uso de las soluciones ofrecidas por la empresa. En castellano este término podría traducirse como *estratega para el éxito*.

2. Objetivos

2.1. Objetivo general

El objetivo general de este proyecto se encuentra enmarcado en, mediante la simulación de una empresa ficticia de servicios IT, adquirir experiencia práctica en el mantenimiento de infraestructuras y servicios. Se busca asegurar la "continuidad del negocio" y la completa disponibilidad de los servicios prestados.

Se perseguirá la comparación entre soluciones *Out-of-the-box* y manuales. En tanto a las **soluciones manuales**, el objetivo es aplicar lo aprendido durante estos dos últimos cursos para poner en funcionamiento los sistemas que darán servicio a los clientes ficticios. El módulo que más se utilizará para este sentido será el de Servicios de Red e Internet, pero esto no implica que sea el único. También se usarán los conocimientos adquiridos en el resto de módulos. Para asegurar los servidores y la continuidad de los servicios y datos, Seguridad y Alta Disponibilidad; para administrar los sistemas operativos y aplicaciones de uso general, Administración de Sistemas Operativos; gestionar las bases de datos (tanto de clientes como las destinadas a uso interno: para los motores DNS y métricas, entre otros), Administración de Sistemas Gestores de Bases de Datos; la creación de interfaces web seguras y útiles para el aprovisionamiento de los clientes y la propia página web de la empresa ficticia de alojamiento web, Implantación de Aplicaciones Web.

Respecto a las **soluciones *Out-of-the-box***, la idea en este sentido es ejemplificar la relativa facilidad de poner en funcionamiento una empresa así, utilizando soluciones ya existentes creadas por empresas desarrolladoras especializadas.

Se han establecido *convenios/patrocinios*[2] con las empresas Axarnet[3] y Clouding.io[4], que ofrecerán sin coste un servidor y crédito para la creación de máquinas virtuales, respectivamente.

[2] Los acuerdos son no vinculantes para con el Instituto. Se realizan a título personal por el alumno.
[3] http://www.axarnet.es/
[4] https://clouding.io/

2.2. Objetivos específicos

Objetivo	Software a utilizar	Componente (Jira)
Disponer de un sistema de VoIP de telefonía que permita mantener la comunicación empresa-cliente. El sistema de telefonía se configurará de forma segura. Se configurarán menús automáticos, buzones de voz y teléfonos VoIP virtuales.	Asterisk, Twilio. Si bien Twilio es una gran empresa, en la cual confían grandes proveedores de servicios de telefonía IP para basar sus servicios, se priorizarán las soluciones nacionales (véase netelip) y europeas.	VoIP
Se ofrecerá un sistema DNS a los clientes. Se estudiará la posibilidad de disponer de planes con únicamente servicio DNS contratado. Será necesario tener visibilidad sobre el tráfico DNS. Se estudiará la posibilidad de disponer de GUI para que los clientes gestionen sus entradas.	Se valorará BIND, atomiaDNS y PowerDNS entre otras opciones teniendo en cuenta las necesidades y objetivos. Collecd o similares, después de un estudio detallado para la recolección de estadísticas. Se utilizarán monitorizadores de estado 53/UDP para comprobar el correcto funcionamiento del/los servidor/es.	Servicio DNS

Se ofrecerán sistemas de alojamientos estático a los clientes. El alojamiento estático, si bien pudiera parecer que es algo propio de los inicios de Internet, está en auge. Una página web estática descarta las complejidades innecesarias, como las consultas a bases de datos, complicación de lado de servidor (PHP). El resultado HTML de la web se genera una única vez y este es servido de forma continuada para todas las peticiones. Por tanto, se aumenta la velocidad de la web, así como la seguridad al no existir bases de datos que se puedan explotar[5]	Se utilizará el servidor web de Apache para servir los sitios web de los clientes. Docker/LXD en el caso de decidir el alojamiento usando contenedores. Si se diera el caso se utilizaría Traefik como servidor web para el balanceo y enrutado hacia contenedores..	Hosting system
Ofrecer servicios de alojamiento dinámico a clientes (Wordpress, OpenCart, Joomla y similares)	Se prevé el uso de contenedores para aislar instancias de clientes que contengan todos los servicios (como Bases de datos) que el cliente necesite.	Hosting system
Disponer de un servicio SMTP y relacionados con email. Al menos de uso interno aunque se estudiará la	Postfix, SpamAssassin, Roundcube.	SMTP & email

[5] 8 Best Static Website Hosting for Business and Personal Use - https://geekflare.com/best-static-site-hosting-platform/

posibilidad de proveer servicios de email a clientes, como paquete combinado en otras ofertas o stand-alone		
Poner a disposición de los clientes un sistema de ticketing mediante el cual puedan iniciar solicitudes (se valorará la creación mediante envío de email) y conocer el estado e histórico de las creadas de forma previa. Los agentes dispondrán de una interfaz para la gestión, respuesta y resolución de los mismos.	Valorar soluciones open-source hosteables: - helpy.io - osticket.com - request - today y similares	Ticketing system
Diseño e implementación de un sistema de backup y recuperación ante desastres mediante el uso de instantáneas geográficamente distribuidas y tareas programadas.	Cron jobs y Shell scripts Google Cloud Storage Instantáneas de Google Compute Engine	GCP manage Security & Backups Hosting system
Posibilidad de acceder a estadísticas y métricas, así como alertas de estado, que permitan a los administradores y administradoras del sistema (valorar	Collecd o similares, después de un estudio detallado para la recolección de estadísticas.	Analytics & dashboards

visibilidad del usuario) conocer el estado y tener visibilidad sobre el sistema. En el caso de fallos o interrupciones del servicio (teniendo en cuenta los SLOs y SLAs con los clientes) será requisito disponer de vías de notificación.	Google Operations y/o Grafana (valorar cloud/hosted) para dashboards y alertas (con sus respectivos agentes de monitorización/recolección de logs). Se valorarán las versiones cloud de estas herramientas para evitar fallos no notificados debido a caídas generalizadas del sistema. Para el seguimiento y análisis web se ofrecerá Matomo Analytics hosted como complemento a los clientes	Security & Backups
Disponer de páginas web públicas para nuevos clientes en la que se puedan conocer (al menos) los servicios ofertados y los precios.	Se estudiará la posibilidad de crear páginas web estáticas o utilizar CMSs para la gestión de los contenidos de la misma.	UX & UI (client side)
(De gestión de proyecto) Trazabilidad, seguimiento y organización de tareas en sistema Kanban	Atlassian Jira Cloud	Enlace Dashboard

2.3. Medios a utilizar

Se utilizarán las aplicaciones y servicios indicados en la sección anterior.

Si bien el software que se utilizará no está completamente definido y puede verse sujeto a modificaciones causadas por, entre otros motivos, mejoras recientes en software descartado, facilidad de implementación, demanda de hardware y funcionalidades de interconexión entre aplicaciones.

En lo relativo a la **gestión del proyecto** se utilizará Atlassian Jira Cloud para el seguimiento de tareas y organización del desarrollo. Esta aplicación/servicio permite planificar, supervisar y gestionar proyectos. Está especialmente diseñado para equipos de desarrollo que usan metodologías ágiles (SCRUM, por ejemplo). Sin embargo, en los últimos años el sistema se ha renovado, permitiendo la gestión de proyectos de todo tipo[6].

Se utilizará Google Drive y su suite de aplicaciones ofimáticas para la confección de **documentación**, así como para el alojamiento de archivos.

Para el alojamiento de archivos de *script* y **control de versiones** se utilizará Github.

Se dispondrá de máquinas virtuales alojadas en Google Cloud Platform. Se valorará la utilización de servicios de operaciones (Operation Suite) ofrecidos por GCP. Si fuera necesario el uso de otras plataformas de *Cloud Computing* se valorará el uso de OVH (solución europea) y clouding.io.

Diferentes **dominios** de internet, tanto para alojar la infraestructura como para simular clientes. Estos dominios estarán registrados con diferentes proveedores y en su mayoría los DNS autoritativos de estos dominios estarán en Cloudflare, salvo excepciones. Se pueden consultar los dominios utilizados en el Anexo I de este documento. Se valorarán otros servicios de esta empresa, Cloudflare, en especial los productos de seguridad, *caching* y ***Zero Trust***[7] [8].

[6] Página principal de Atlassian Jira - https://www.atlassian.com/es/software/jira
[7] Introducción a Cloudflare Zero Trust - https://www.cloudflare.com/es-es/products/zero-trust/zero-trust-network-access/
[8] ¿Qué es la Seguridad Zero Trust? - https://www.cloudflare.com/es-es/learning/security/glossary/what-is-zero-trust/

3. Actividades y tareas - Métodos y técnicas.

Todas las actividades y tareas de este Proyecto Fin de Ciclo del Ciclo Formativo de Grado Superior en Administración de Sistemas Informáticos en Red estarán relacionadas con la Administración de Sistemas, el despliegue y la operación continuada de los servicios ofrecidos.

La metodología de trabajo para este proyecto se basará en la instalación de los propios servicios, a la vez que se documenta todo el proceso de instalación y configuración de sistemas operativos y aplicaciones.

Respecto a los ajustes aplicados a las aplicaciones, puesto que la mayoría serán instalados en sistemas sin interfaz gráfica (mayormente sistemas operativos GNU/Linux), se mostrarán en este documento los fragmentos modificados que puedan ser de interés.

Véase el siguiente ejemplo:

/etc/apache2/sites-available

```
<VirtualHost *:80>
        ServerAdmin support@glez.cloud
        DocumentRoot /var/www/k8s
        ServerName k8s.gonzaleztroyano.es
        DirectoryIndex k8s.pdf
        ErrorLog /var/log/apache2/k8s-error.log
        CustomLog /var/log/apache2/access.log combined
RewriteEngine on
RewriteCond %{SERVER_NAME} =d-k8s.gonzaleztroyano.es
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
[END,NE,R=permanent]
</VirtualHost>
```

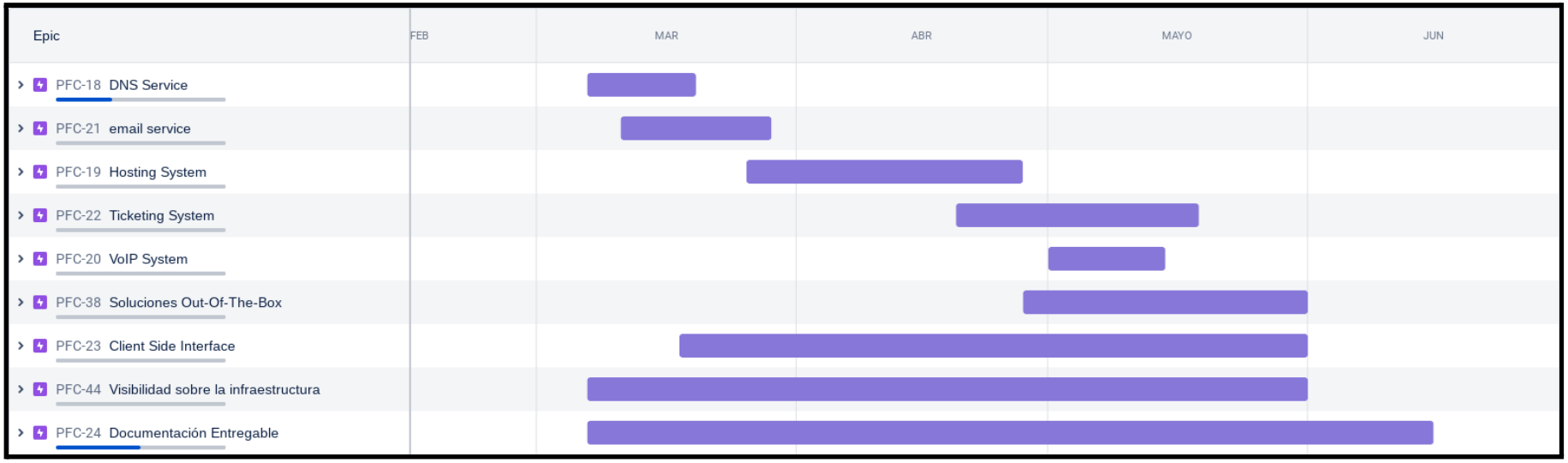
También se almacenarán en un repositorio en Github[9] creado específicamente para este proyecto. En el caso de que sea relevante para la demostración del correcto funcionamiento de un sistema, servicio o aplicación dado, también se podrán grabar vídeos y la terminal de comandos. Para la publicación de los videos se utilizará uno plataforma específica para este fin, Loom[10]. En tanto a la grabación y publicación de las sesiones de terminal, se utilizará otra plataforma específica para esta tarea, Asciinema[11]. Como ejemplo, se adjunta a esta documentación esta grabación de terminal[12] realizada para ejemplificar el funcionamiento de un *script* del módulo de Implantación de Aplicaciones Web y esta grabación[13] de la pantalla realizada para el proyecto del primer trimestre del módulo de Seguridad y Alta Disponibilidad.

Toda la documentación, configuración, imágenes, vídeos y demás entregables serán puestos a disposición de cualquier persona en Internet.

[9] Repositorio de Github del Proyecto - https://github.com/gonzaleztroyano/ASIR2-PFC/
[10] Página principal de la herramienta Loom - https://www.loom.com/
[11] Página principal de la herramienta asciinema - https://asciinema.org/
[12] Ejemplo de grabación de terminal con asciinema realizada para el módulo de Implantación de Aplicaciones Web - https://asciinema.org/a/451200
[13] Ejemplo de grabación con Loom - https://www.loom.com/share/2ca8a17c02a64444b19793560afb7d63

4. Cronograma

Como se ha mencionado anteriormente en este documento, se utilizará la aplicación Jira, de Atlassian, para gestionar el trabajo durante el proyecto. Al comienzo del proyecto (inicios de marzo), el panel de control de tareas está de la siguiente manera:



La imagen superior muestra los “epics” del proyecto. Un *epic* se puede entender como *un conjunto de trabajo grande que puede dividirse en tareas específicas*[14]. Temporalmente, se organizará mediante *sprints*. Un *sprint* es *un período breve de tiempo fijo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida*[15]. Los sprints tendrán una duración de entre una y dos semanas.

[14] Definición de *epic* en la página web de Atlassian - https://www.atlassian.com/es/agile/project-management/epics
[15] Definición de *sprint* en la página web de Atlassian - https://www.atlassian.com/es/agile/scrum/sprints

5. Soluciones *manuales*

5.1. Servicio DNS 📖

5.1.1 Introducción al servicio

Como bien hemos visto en el módulo de Servicios de Red e Internet y tal y como podemos leer en este artículo de Cloudflare[16], el servicio de nombres de dominio (*Domain Name Service;* en adelante, *DNS*) es el listín telefónico de Internet.

Como *overview* técnico, podemos decir que el servicios DNS convierte nombre de dominio (véase “madrid.org” o “gonzaleztroyano.es”) a direcciones IP.

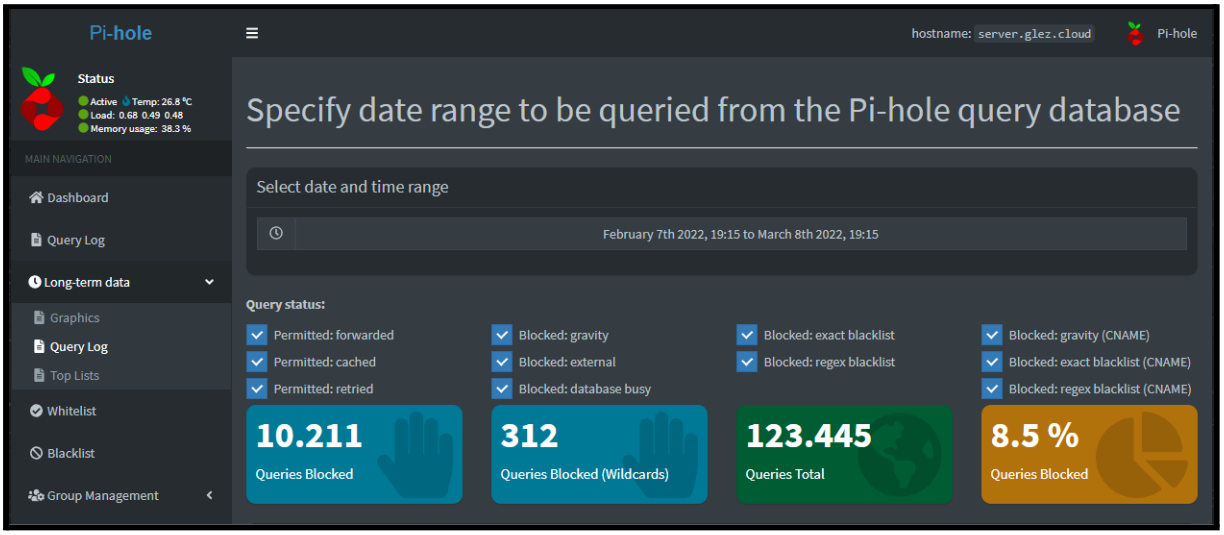
[17]

Pero esta función es únicamente de los registros A y AAAA (o hasta incluso CNAME). En un primer momento, el RFC 1035[18] definió 14 tipos de “Recursos de Registro”, RR o Registros para abreviar. Hoy en día la lista de actualizaciones es extensa: RFC 1101, RFC 1183, RFC 1348, RFC 1876, RFC 1982, RFC 1995, RFC 1996, RFC 2065, RFC 2136, RFC 2181, RFC 2137, RFC 2308, RFC 2535, RFC 2673, RFC 2845, RFC 3425, RFC 3658, RFC 4033, RFC 4034, RFC 4035, RFC 4343, RFC 5936, RFC 5966, RFC 6604, RFC 7766, RFC 8482, RFC 8490, RFC 8767. Todos los RFC anteriores mejoran, actualizan o mejoran el primer estándar

[16] Artículo de Cloudflare explicando el concepto de DNS - https://www.cloudflare.com/es-es/learning/dns/what-is-dns/
[17] Imagen obtenida del libro *Managing Mission-Critical Domains and DNS*. Autor Mark E. Jeftovic. Consultado en marzo de 2022
[18] Request For Coments número 1035 - Disponible aquí: https://www.rfc-editor.org/rfc/rfc1035.html

del DNS. De forma adicional, la IANA (*Internet Assigned Numbers Authority*) mantiene y actualiza esta lista[19] con todas las clases de DNS, los códigos de opciones, los registros de recursos, opciones de cabeceras, etc.

No hay duda de que el servicio DNS es vital. Como ejemplo, en mi casa (donde el uso de internet es limitado y completamente típico para un hogar español con 4 miembros) se han generado más de 120.000 consultas DNS en 30 días. Esta cifra supone una media de más de 4.000 consultas DNS al día.



Tal y como se describe en el anteproyecto y en el desglose de objetivos específicos, para el servicio DNS se debe:

> *Ofrecer servicios DNS a clientes*
>
> *Se estudiará la posibilidad de disponer de planes con únicamente servicio DNS contratado*
>
> *Será necesario tener visibilidad sobre el tráfico DNS*
>
> *Se estudiará la posibilidad de disponer de GUI para que los clientes gestionen sus entradas.*

[19] Lista de parámetros DNS de la IANA - https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml

5.1.2. Instalación de PowerDNS

Como servidor DNS se ha elegido PowerDNS[20] sobre BIND[21] puesto que es la solución elegida por los grandes proveedores de servicios como plataforma. Si bien a nuestro nivel el uso va a ser limitado (e incluso el uso de *dnsmasq* como servidor hubiera sido más que suficiente[22]), la idea del Proyecto es simular al máximo las condiciones de configuración para una empresa de alojamiento web.

5.1.2.1. Creación de la máquina

Como se ha comentado anteriormente, la práctica mayoría de los servicios y máquinas virtuales estarán alojadas en Google Cloud Platform. Es probable que a lo largo del proyecto se cambie de plataforma.

Lanzaremos una máquina virtual para alojar el servicio. Lo más cómodo es ejecutar el siguiente comando **gcloud**[23]:

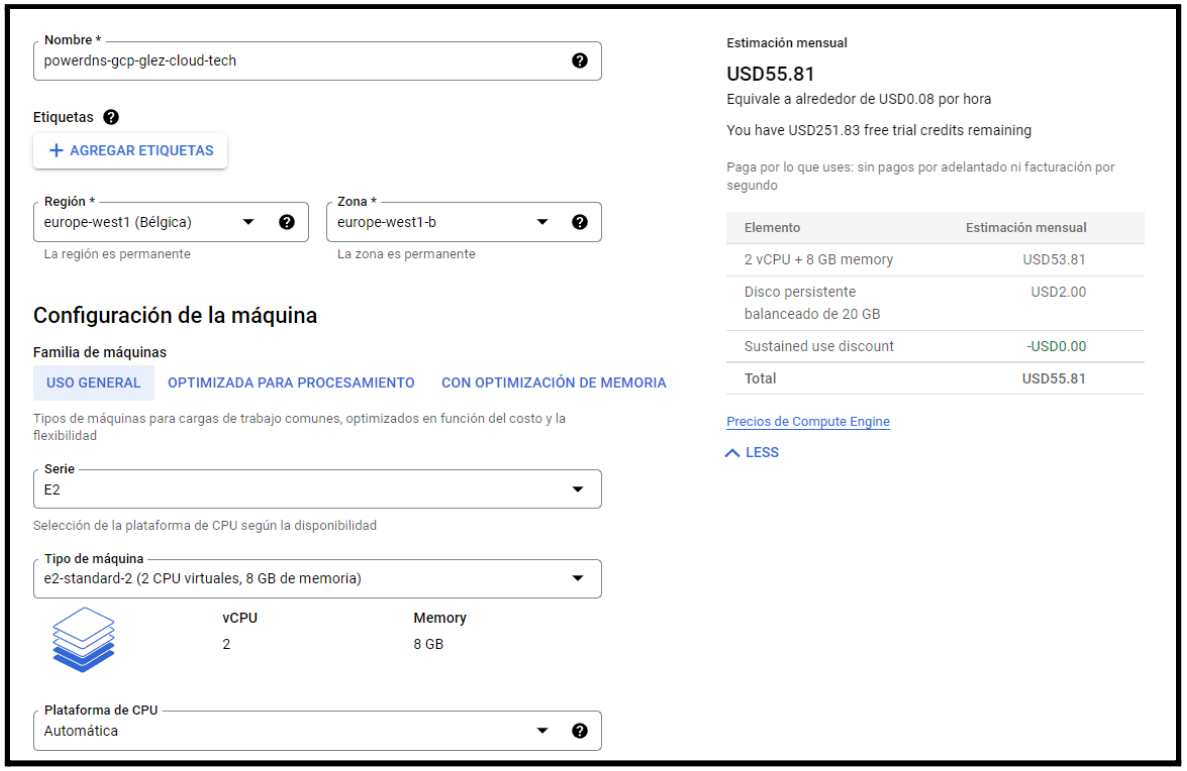
```
gcloud compute instances create powerdns-gcp-glez-cloud-tech
--project=gcp-test-pablo-glez-asir2 --zone=europe-west1-b
--machine-type=e2-standard-2
--network-interface=network-tier=PREMIUM,subnet=default
--maintenance-policy=MIGRATE
--service-account=488992079897-compute@developer.gserviceaccount.com
--scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www
.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monito
ring.write,https://www.googleapis.com/auth/servicecontrol,https://www.goo
gleapis.com/auth/service.management.readonly,https://www.googleapis.com/a
uth/trace.append
--tags=portainer,powerdns-public,powerdns-admin,http-server,https-server
--create-disk=auto-delete=yes,boot=yes,device-name=powerdns-gcp-glez-clou
d-tech,image=projects/ubuntu-os-cloud/global/images/ubuntu-1804-bionic-v2
0220302,mode=rw,size=20,type=projects/gcp-test-pablo-glez-asir2/zones/eur
ope-west1-b/diskTypes/pd-balanced --no-shielded-secure-boot
--shielded-vtpm --shielded-integrity-monitoring
--reservation-affinity=any
```

[20] Página principal de PowerDNS - https://www.powerdns.com/
[21] Página principal de BIND - https://www.isc.org/software/bind
[22] Artículo en la wikipedia sobre dnsmasq - https://es.wikipedia.org/wiki/Dnsmasq
[23] Referencia de comando gcloud - developers.google.com/cloud/sdk/gcloud/reference/compute/

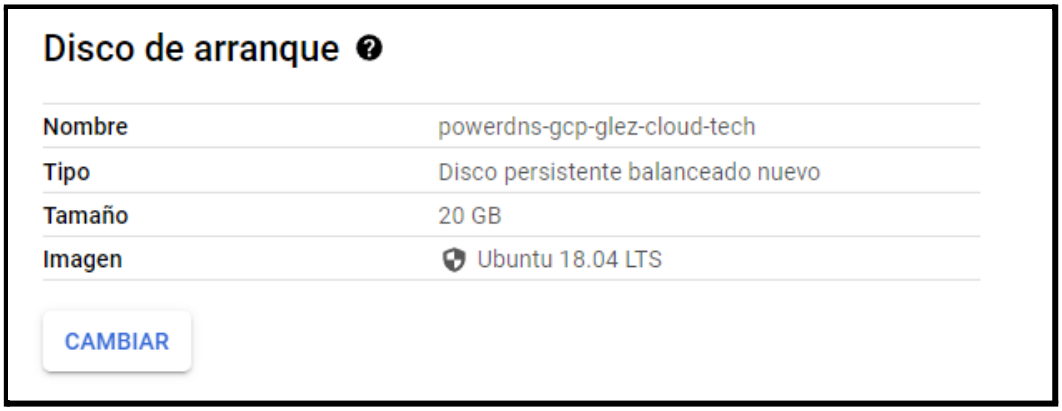
También podemos utilizar la **GUI web** para crearla:

Definimos el nombre de la máquina, la región y zona donde se creará. En nuestro caso, Bélgica.

Al seleccionar el tipo de máquina (*e2-standard-2*, en nuestro caso) automáticamente se actualizará la estimación de coste mensual.



Como disco de arranque seleccionamos un Ubuntu 18.04 LTS de 20 GB de tipo persistente balanceado. Dependiendo del rendimiento esperado, podemos elegir otro tipo de disco y añadir varios, hasta virtualmente cualquier capacidad.



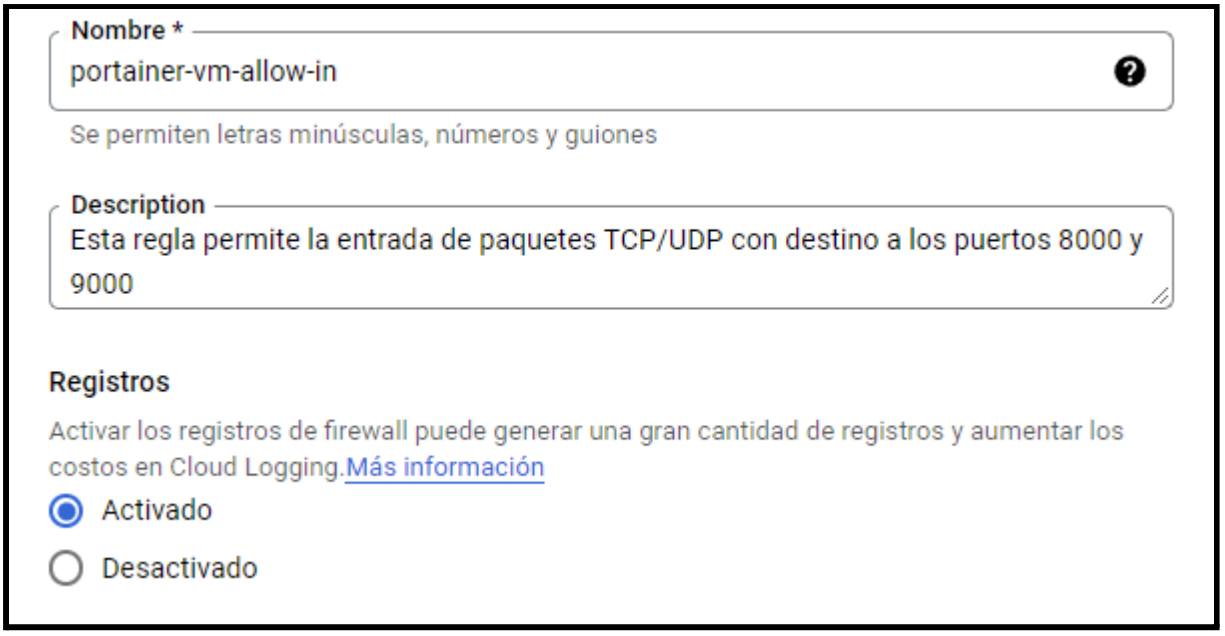
El firewall en Google Cloud Platform puede gestionarse mediante etiquetas de red[24]. La aplicación de estas etiquetas hace sencilla la administración de la seguridad de la red. Estas etiquetas también son aplicables, por ejemplo, a los balanceadores de carga administrados de Google Cloud.

5.1.2.2. Crear reglas de firewall

Además de aplicar las etiquetas de red correspondientes, debemos indicar a Google Cloud qué reglas relacionan a estas etiquetas. Para hacerlo navegamos, dentro de la interfaz web de GCP hasta *Red de VPC* > *Firewall*. Aquí hacemos clic en *Crear regla de Firewall*. Veamos un ejemplo:

Definimos el nombre de la regla (recordemos que en Google Cloud Platform, y en virtualmente cualquier nube pública, todo es un recurso). Es recomendable añadir una descripción para poder identificarla de forma sencilla posteriormente. Más aún si se está trabajando con más personas en un mismo proyecto

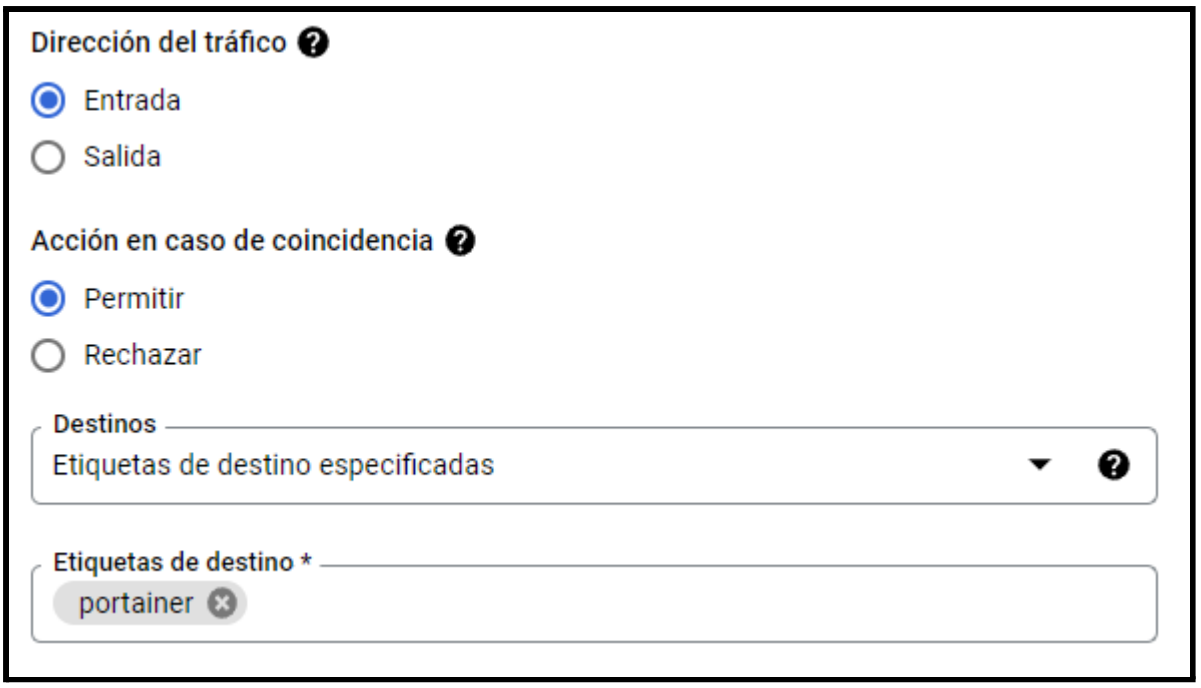
[24] Gestión de etiquetas de red en las VM de GCP - https://cloud.google.com/vpc/docs/add-remove-network-tags

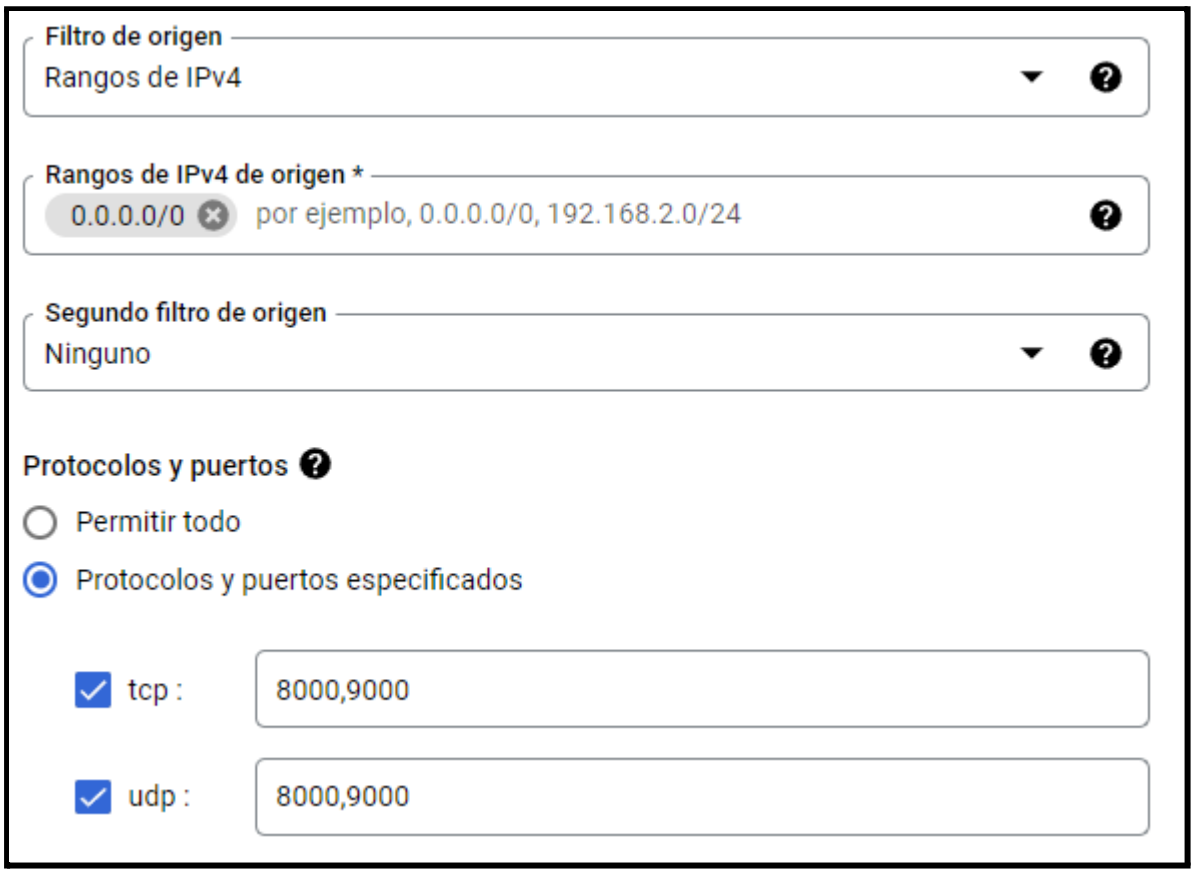


Activamos los registros (incluyendo los metadatos), pues nos darán una gran información sobre el número de solicitudes de acceso a la VM (y su decisión de enrutamiento).

La red a la que aplicaremos será *default*. Como dato, en GCP podemos crear redes virtuales que se pueden extender a una sola región o a todo el planeta. De esta forma, podemos tener cada servicio dentro de una red distintas e intercomunicarlos posteriormente si fuera necesario.

Las reglas en GCP son muy parecidas a las que hemos visto sobre iptables en el módulo de Seguridad y Alta Disponibilidad, como podemos ver en las siguientes imágenes en las que las definimos:





Para crear el esta y el resto de redes también podemos ejecutar en Cloud Shell los siguientes comandos:

```
gcloud compute --project=gcp-test-pablo-glez-asir2 firewall-rules create
portainer-vm-allow-in --description="Esta regla permite la entrada de
paquetes TCP/UDP con destino a los puertos 8000 y 9000"
--direction=INGRESS --priority=1000 --network=default --action=ALLOW
--rules=tcp:8000,tcp:9000,udp:8000,udp:9000 --source-ranges=0.0.0.0/0
--target-tags=portainer --enable-logging
```

```
gcloud compute --project=gcp-test-pablo-glez-asir2 firewall-rules create
powerdns-public-allow-in --description="Esta regla permite la entrada de
paquetes TCP/UDP con destino al puerto 53" --direction=INGRESS
--priority=1000 --network=default --action=ALLOW --rules=tcp:53,udp:53
--source-ranges=0.0.0.0/0 --target-tags=powerdns-public --enable-logging
```

```
gcloud compute --project=gcp-test-pablo-glez-asir2 firewall-rules create
powerdns-admin-allow-in --description="Esta regla permite la entrada de
paquetes TCP/UDP con destino al puerto 53" --direction=INGRESS
--priority=1000 --network=default --action=ALLOW
--rules=tcp:8081,tcp:9191 --source-ranges=0.0.0.0/0
--target-tags=powerdns-admin --enable-logging
```

Si en la web GUI hemos seleccionado “Permitir tráfico HTTP” y/o “Permitir tráfico HTTPS” automáticamente se añaden otras etiquetas de red. `http-server` y `https-server`, respectivamente.

5.1.2.3. Configuración de la máquina e instalación de paquetes

Procederemos a continuación a configurar los paquetes necesarios para que la VM pueda convertirse en un servidor DNS.

Antes de nada, actualizaremos los repositorios y aplicaremos las actualizaciones que la máquina tuviera pendientes. También instalaremos propio powerdns y el backend de sqlite, donde se almacenarán los registros:

```
sudo apt update -y && sudo apt upgrade -y
sudo apt install pdns-server -y
sudo apt install pdns-backend-sqlite3 -y
```

PowerDNS es capaz de utilizar distintos *backends* donde se almacenarán los registros. Desde simples archivos de texto en formato BIND, hasta bases de datos

MySQL hasta archivos sqlite como el que vamos a utilizar debido a la sencillez de gestión.

Guardamos el archivo pdns.conf con otro nombre y mandamos a este las configuraciones para que se apliquen:

```
sudo mv /etc/powerdns/pdns.conf /etc/powerdns/pdns.conf.bak
sudo touch /etc/powerdns/pdns.conf
echo -e
"#######\napi=yes\napi-key=JmnWB4iiphR6FyzygJ3sdrx1u5OCas\napi-logfile=/
var/log/pdns.log\n#######\nwebserver=yes\nwebserver-address=0.0.0.0\nweb
server-allow-from=0.0.0.0/0,127.0.0.1\nwebserver-port=8081\n#######\nset
gid=pdns\nsetuid=pdns\n#######\nlaunch=gsqlite3\ngsqlite3-database=/var/
lib/powerdns/pdns.sqlite3\n" > /etc/powerdns/pdns.conf
```

5.1.2.4. Creación de la BDD

Creamos la base de datos sqlite3 donde se almacenarán los registros y que gestionará PowerDNS:

```
sudo mkdir /var/lib/powerdns
wget
https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-PFC/main/1-dns/s
chema.sqlite3.sql
sudo sqlite3 /var/lib/powerdns/pdns.sqlite3 < schema.sqlite3.sql
sudo chown -R pdns:pdns /var/lib/powerdns
```

Este archivo que estamos utilizando como base para el esqueleto de la base de datos es un archivo SQL con órdenes de creación de tablas (CREATE TABLES). Es muy sencillo, el archivo creado está disponible en Github[25] y en el apartado 7.4.1 de este documento

5.1.2.5. Liberar el puerto 53/UDP

Como sabemos de nuestros apuntes del módulo de Servicios de Red e Internet, el servidor DNS escucha en el puerto 53, tanto en UDP (de forma predeterminada) como en TCP (en versiones más modernas y seguras de protocolo).

[25] Archivo en Github - https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/1-dns/schema.sqlite3.sql

Por defecto Ubuntu, y muchos sistemas Linux, tienen su propio servidor DNS interno funcionando. Debemos desactivarlo para permitir que PowerDNS utilice el puerto 53/UDP.

```
sudo systemctl disable systemd-resolved
sudo systemctl stop systemd-resolved
```

Eliminamos el enlace simbólico (si lo tuviera) del archivo `/etc/resolv.conf` y renombramos el archivo por si hubiera que recuperarlo. Creamos el nuevo archivo.

```
ls -lh /etc/resolv.conf  # Para ver si existiera el enlace
sudo mv /etc/resolv.conf /etc/resolv.conf.bak
echo "nameserver 1.1.1.1" | sudo tee /etc/resolv.conf
echo "127.0.0.1 powerdns.gcp.glez-cloud.tech." >> /etc/hosts
echo "127.0.0.1 powerdns" >> /etc/hosts
echo "powerdns.gcp.glez-cloud.tech" > /etc/hostname
```

5.1.2.6. Reinicio y comprobaciones de servicio

Reiniciemos el servicio y habilitémosle para reiniciarse

```
sudo systemctl restart pdns
sudo systemctl enable pdns
```

Podemos usar el comando `sudo systemctl status pdns` para ver el estado del servicio:

```
pablogontroya@powerdns:~$ sudo systemctl status pdns
● pdns.service - PowerDNS Authoritative Server
     Loaded: loaded (/lib/systemd/system/pdns.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-03-09 19:59:22 UTC; 20s ago
       Docs: man:pdns_server(1)
             man:pdns_control(1)
             https://doc.powerdns.com
   Main PID: 5896 (pdns_server)
      Tasks: 10 (limit: 4915)
     CGroup: /system.slice/pdns.service
             └─5896 /usr/sbin/pdns_server --guardian=no --daemon=no --disable-syslog --log-timestamp=no --write-pid=no

Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: TCPv6 server bound to [::]:53
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: PowerDNS Authoritative Server 4.1.1 (C) 2001-2017 PowerDN
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: Using 64-bits mode. Built using gcc 7.3.0.
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: PowerDNS comes with ABSOLUTELY NO WARRANTY. This is free
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: Listening for HTTP requests on 0.0.0.0:8081
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: Could not retrieve security status update for '4.1.1-1.Ub
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: Creating backend connection for TCP
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: About to create 3 backend threads for UDP
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech systemd[1]: Started PowerDNS Authoritative Server.
Mar 09 19:59:22 powerdns.gcp.glez-cloud.tech pdns_server[5896]: Done launching threads, ready to distribute questions
```

En la captura de pantalla anterior podemos ver que el servicio está funcionando correctamente.

También podemos ver si el servidor está escuchando peticiones en el puerto 54 utilizando el siguiente comando:

```
sudo netstat -tap | grep pdns
```

```
pablogontroya@powerdns:~$ sudo netstat -tap | grep pdns
tcp        0      0 0.0.0.0:domain          0.0.0.0:*               LISTEN      5896/pdns_server
tcp        0      0 0.0.0.0:tproxy          0.0.0.0:*               LISTEN      5896/pdns_server
tcp6       0      0 [::]:domain             [::]:*                  LISTEN      5896/pdns_server
```

Con el comando `dig`, incluido en el paquete `dnsutils` podemos "preguntar" a nuestro servidor para ver si responde:

```
sudo apt install dnsutils -y
dig @127.0.0.1
```

```
pablogontroya@powerdns:~$ dig @127.0.0.1

; <<>> DiG 9.11.3-1ubuntu1.16-Ubuntu <<>> @127.0.0.1
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 22173
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1680
;; QUESTION SECTION:
;.                              IN      NS

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Mar 09 20:03:31 UTC 2022
;; MSG SIZE  rcvd: 28
```

5.1.2.7. Añadir datos de prueba y comprobar

Ahora que sabemos que el servidor DNS que acabamos de instalar está funcionando; pero debemos saber si está funcionando correctamente.

Para hacerlo, crearemos una nueva zona con algunos registros y después preguntaremos a nuestro servidor DNS sobre estos.

Podemos generar la zona DNS nueva en nuestro servidor utilizando la utilidad de comandos "`pdnsutil`".

```
sudo su
pdnsutil create-zone pablito.com ns1.example.com
pdnsutil add-record pablito.com '' MX '25 mail.pablo.gonzalez'
pdnsutil add-record pablito.com www A 192.0.2.1
exit
dig +short pablito.com @127.0.0.1
dig +short pablito.com MX @127.0.0.1
```

Los resultados de las consultas serán los mismos, ya se consulten desde la propia máquina o desde cualquier máquina de internet.

Esta es la respuesta desde el ordenador de mi casa:

```
pablo@WIN-PABLO:~$ dig +short pablito.com NS @powerdns.gcp.glez-cloud.tech
ns1.example.com.
pablo@WIN-PABLO:~$ dig +short pablito.com MX @powerdns.gcp.glez-cloud.tech
25 mail.pablo.gonzalez.
pablo@WIN-PABLO:~$ dig pablito.com NS @powerdns.gcp.glez-cloud.tech

; <<>> DiG 9.16.1-Ubuntu <<>> pablito.com NS @powerdns.gcp.glez-cloud.tech
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1562
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1680
;; QUESTION SECTION:
;pablito.com.                   IN      NS

;; ANSWER SECTION:
pablito.com.            3600    IN      NS      ns1.example.com.

;; Query time: 25 msec
;; SERVER: 130.211.111.12#53(130.211.111.12)
;; WHEN: Wed Mar 09 21:21:03 CET 2022
;; MSG SIZE  rcvd: 66

pablo@WIN-PABLO:~$
```

5.1.3. Instalación y pruebas de PowerDNS Admin GUI

5.1.3.1. Instalación de docker Engine y Portainer

La herramienta de gestión DNS vía web, en adelante *PowerDNS Admin*, permite administrar las zonas DNS, los registros de estas, y los accesos a la web de forma sencilla. Esta herramienta se gestionará en un contenedor, para reforzar los conocimientos adquiridos durante este curso en el módulo de Seguridad y Alta Disponibilidad y por la facilidad que representa la ejecución en contenedores. También es posible instalarla mediante instalación nativa en el sistema.

Antes de nada, vamos a instalar el motor de docker siguiendo las instrucciones contenidas en la página oficial[26]:

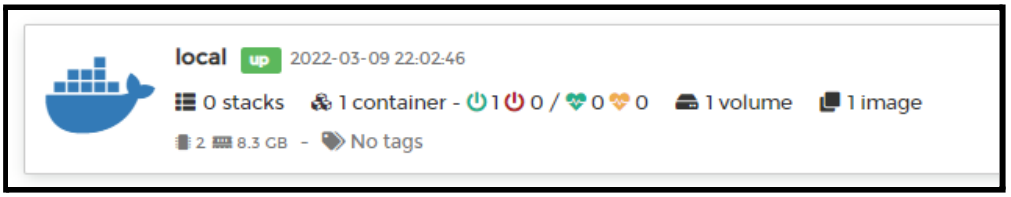
```
sudo apt-get remove docker docker-engine docker.io containerd runc
curl -fsSL https://get.docker.com -o get-docker.sh
chmod +x get-docker.sh
./get-docker.sh
```

Además, instalaremos la herramienta web gráfica de gestión de contenedores Portainer[27]. Como curiosidad, también es un contenedor.

```
sudo docker volume create portainer_data
sudo docker run -d -p 8000:8000 -p 9000:9000 --name=portainer
--restart=always -v /var/run/docker.sock:/var/run/docker.sock -v
portainer_data:/data portainer/portainer-ce
```

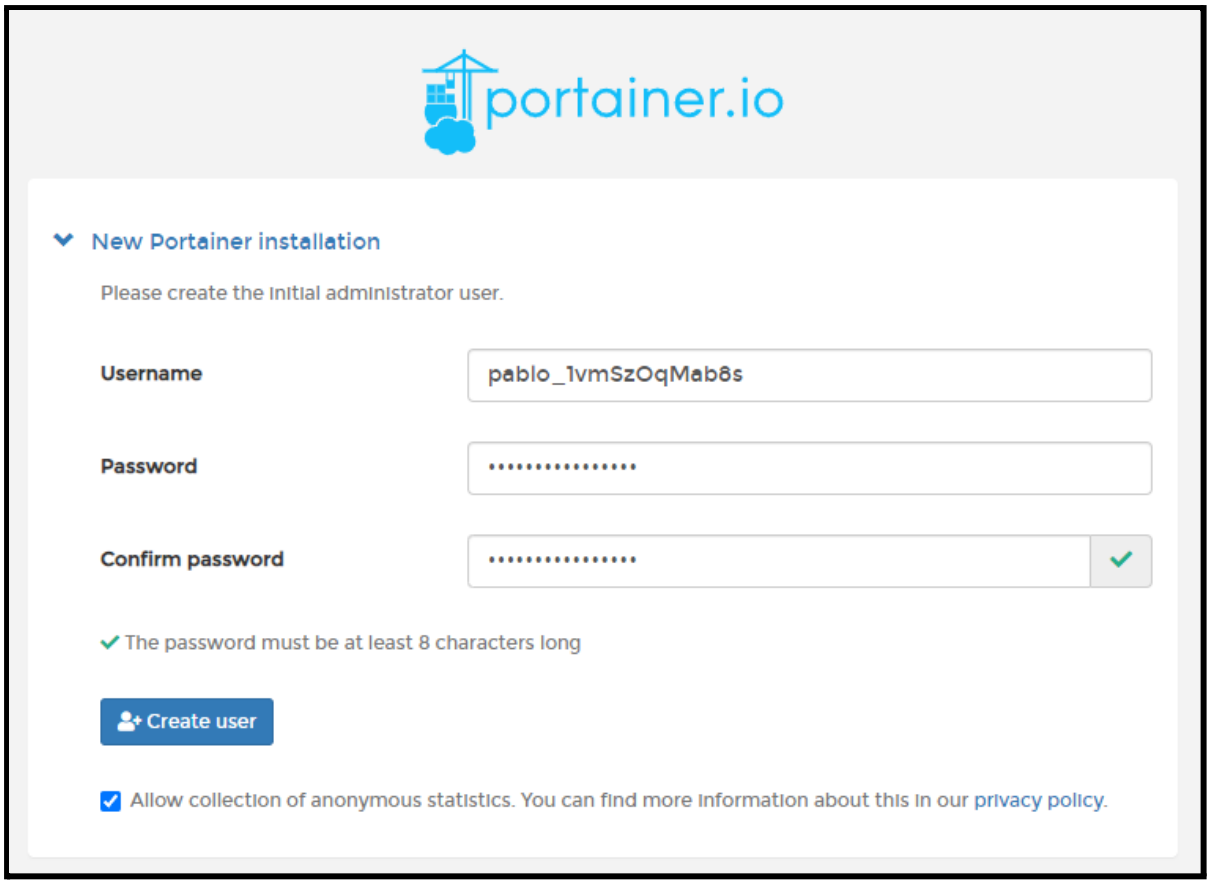
En el primer arranque, nos solicitará que introduzcamos un usuario, que será superadministrador de portainer. También debemos definir su contraseña.

Creamos el usuario haciendo clic en *Create user*. En la pantalla ya veremos el entorno local:



[26] Manual de instalación de instalación de Docker - https://docs.docker.com/engine/install/ubuntu/
[27] Página principal de Portainer - https://www.portainer.io/

Esta es la primera pantalla que vemos al iniciar portainer:



5.1.3.2. Instalar docker-compose y levantar el deployment

Como sabemos, la generación y mantenimiento de *stacks* basados en contenedores es mucho más sencilla si utilizamos estas "recetas" que nos permiten escribir de forma declarativa el entorno que queremos generar en docker. De forma previa a cualquier acción, debemos instalar la herramienta:

```
 sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-compo
se-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
chmod u+x /usr/local/bin/docker-compose
```

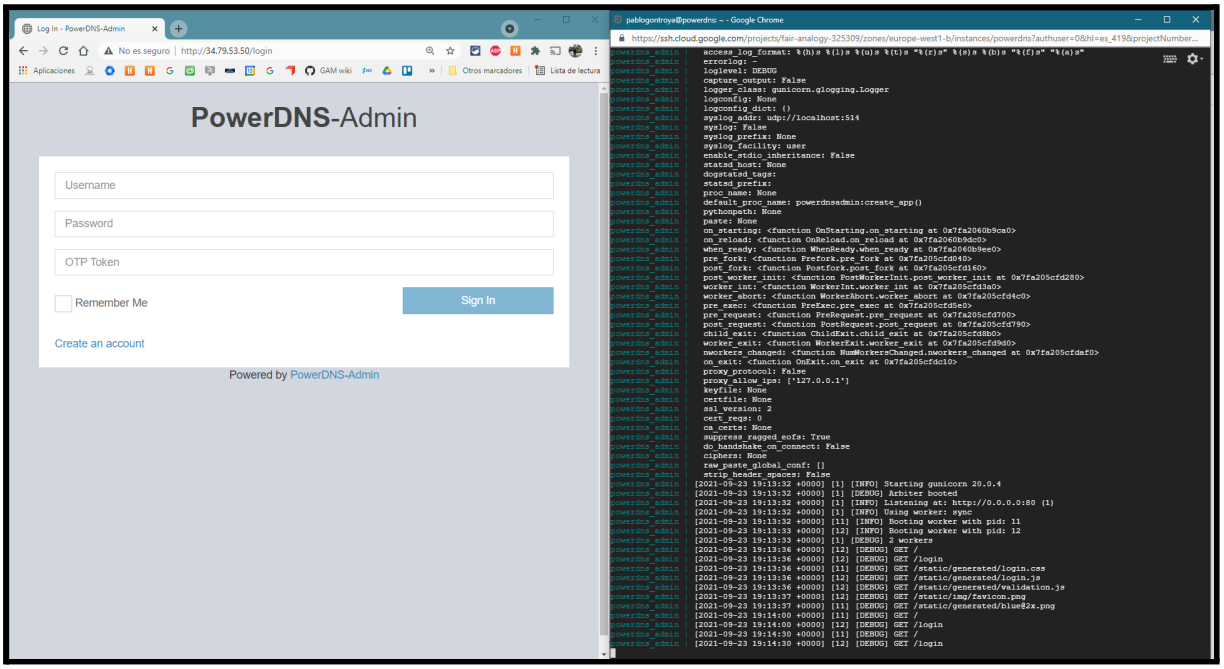
A continuación, vamos a crear una carpeta para el despliegue y descargaremos desde el repositorio de Github el contenido del archivo docker-compose.yml:

```
mkdir powerdns-admin
wget https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-PFC/main/1-dns/docker-compose.yml
docker-compose up
```

```
Creating volume "pablogontroya_pda-data" with default driver
Pulling app (ngoduykhanh/powerdns-admin:latest)...
latest: Pulling from ngoduykhanh/powerdns-admin
5758d4e389a3: Pull complete
2c40eb35ff36: Pull complete
b218723011f1: Pull complete
41c45f2f26c3: Pull complete
7f33384f5962: Pull complete
3e4abb6e835a: Pull complete
4b675c72e18e: Pull complete
Digest: sha256:67886c94cf627b351269313ee3564d8bc9d4a0cef6986a15ba5c34459f1aee16
Status: Downloaded newer image for ngoduykhanh/powerdns-admin:latest
Creating powerdns_admin ... done
root@powerdns:/home/pablogontroya#
```

5.1.3.3. Acceso a PowerDNS-Admin

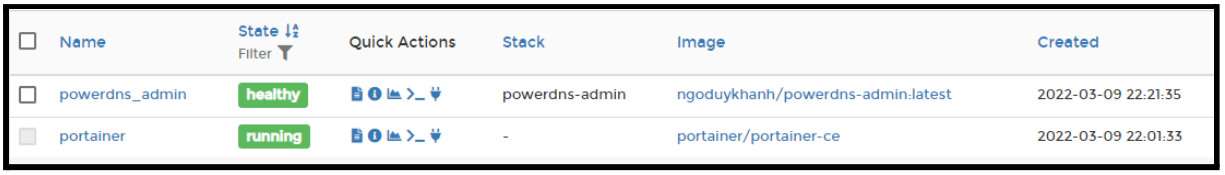
Si ahora accedemos al navegador (mientras mantenemos abierta la terminal de docker compose):



Utilizamos `Ctrl+C` para detener el proceso. Podemos gestionarlo desde portainer, donde ya lo tendremos registrado. Lo vemos parado pues lo hemos parado desde la línea de comandos. Vamos a realizar las siguientes acciones:

- En “RESTART POLICIES“ vamos a definir Unless Stopped, haciendo clic en “Update” para aplicar los cambios.
- Hacemos clic en el botón de Start, en la parte superior de la página del contenedor.

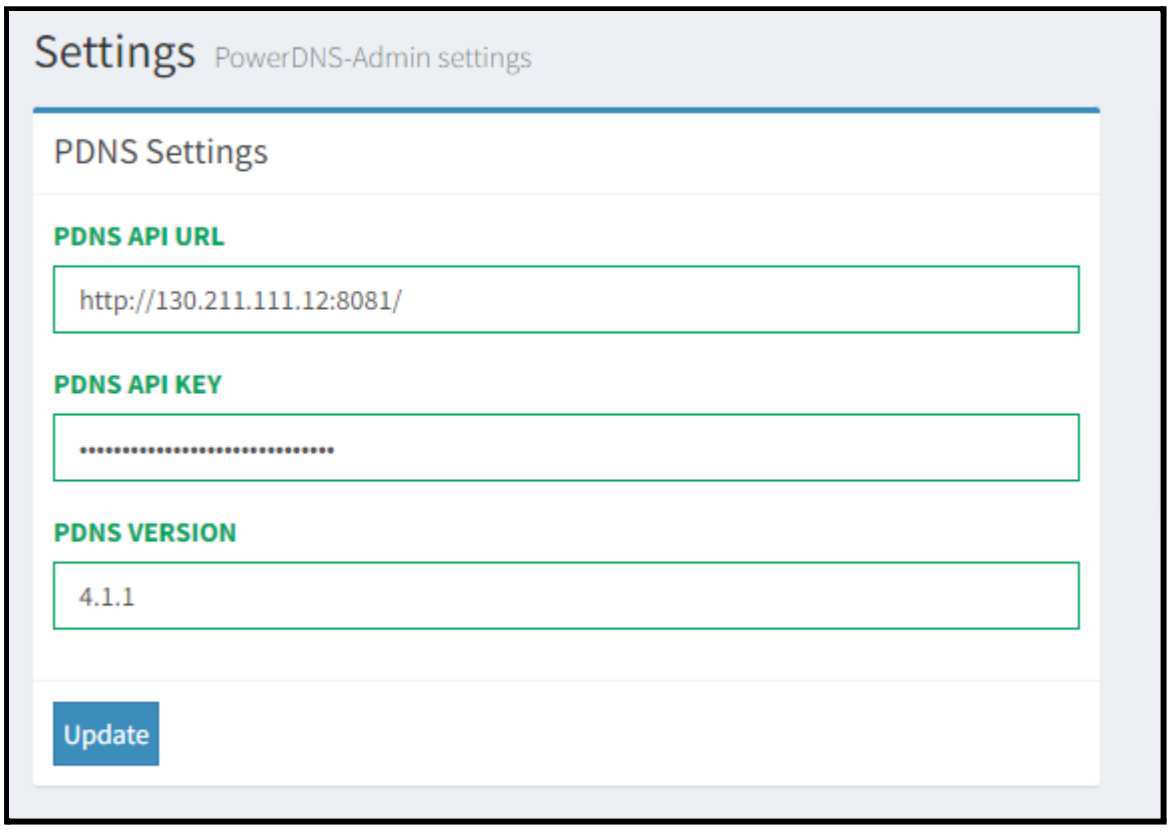
Así es como veremos la pantalla de gestión de contenedores después de iniciarlo:



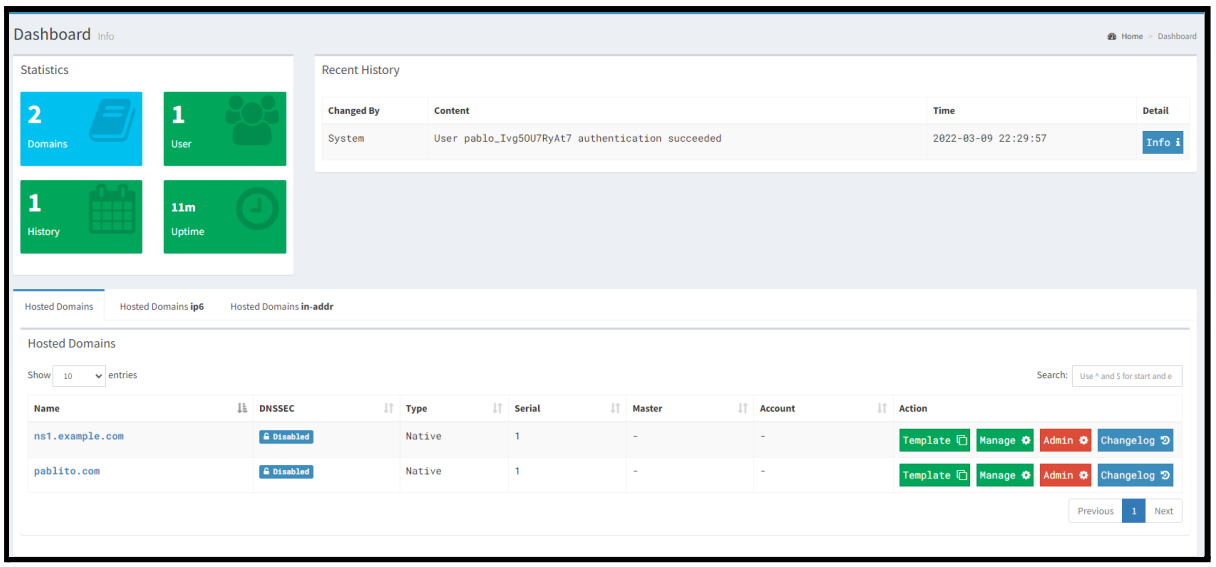
Volviendo a PowerDNS Admin, el primer usuario que se registre haciendo clic en "*Create an account*" será el administrador.

5.1.3.4. Conexión de PowerDNS Admin con la API

La interfaz gráfica utiliza la API de PowerDNS para aplicar las acciones. Debemos introducir la IP y el puerto de conexión (http://10.132.0.22:8081) definido en pdns.conf, así como la contraseña.

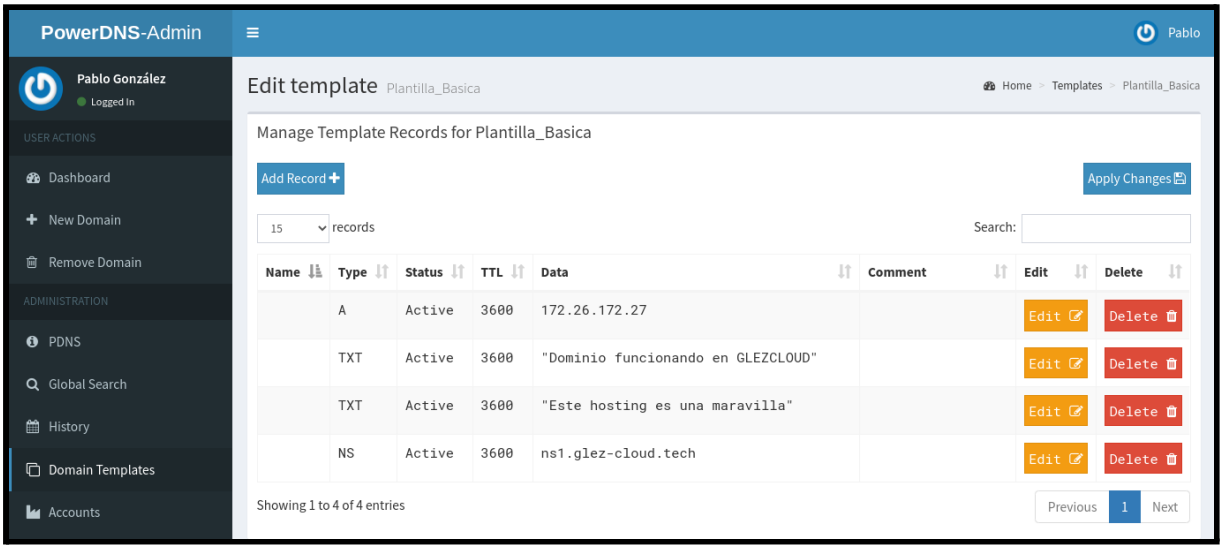


Ahora, si navegamos hasta el dashboard ya podemos ver tanto la cuenta de los dominios como la lista de estos:



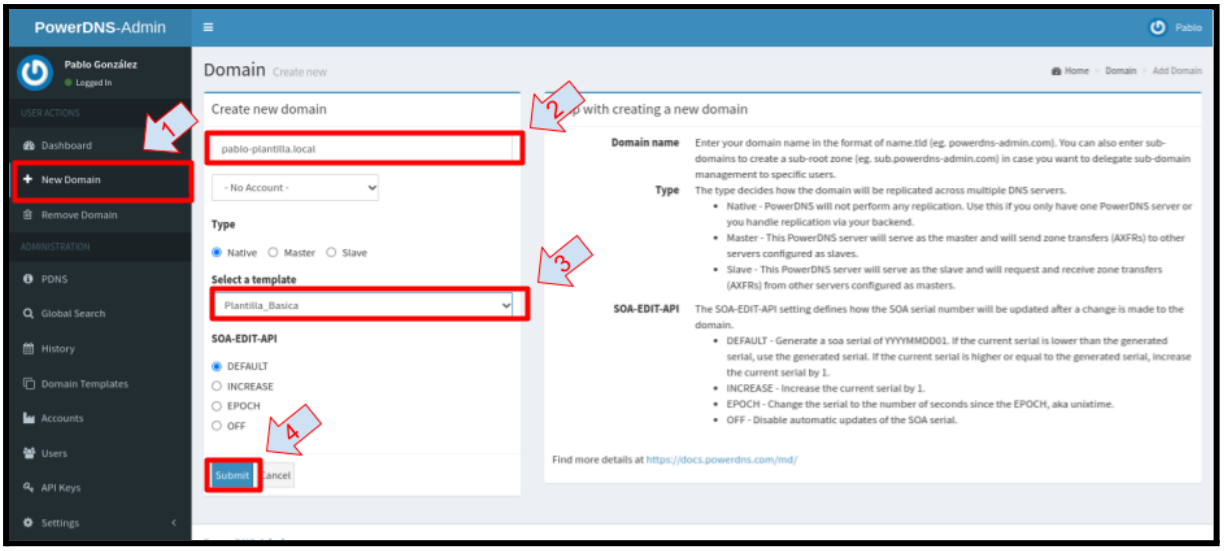
5.1.3.5. Plantillas de zona y cuentas

Antes de continuar con la creación de cuentas y dominios en el servidor DNS, se va a crear una plantilla, que será la que aplicaremos a los nuevos dominios en su creación. Esta plantilla tiene el siguiente contenido:



Una vez hemos creado la plantilla, lo que vamos a hacer es crear otro dominio de prueba para comprobar que se están aplicando correctamente las zonas de prueba.

Para crear una nueva zona, simplemente debemos hacer clic en “New Domain”. Después, introducimos los datos solicitados y hacemos clic en “Submit” para proceder a crear el dominio.



5.1.4. Activación de logs

Por defecto, powerdns solo registra los errores graves[28]. Debemos cambiar este ajuste para que registremos los detalles de las consultas.

Ejecutamos el siguiente comando:

```
sudo systemctl edit --full pdns.service
```

En el archivo temporal que se nos abre, retiramos "`--guardian=no`"de la línea:

```
ExecStart=/usr/sbin/pdns_server --guardian=no --daemon=no
--disable-syslog --log-timestamp=no --write-pid=no
```

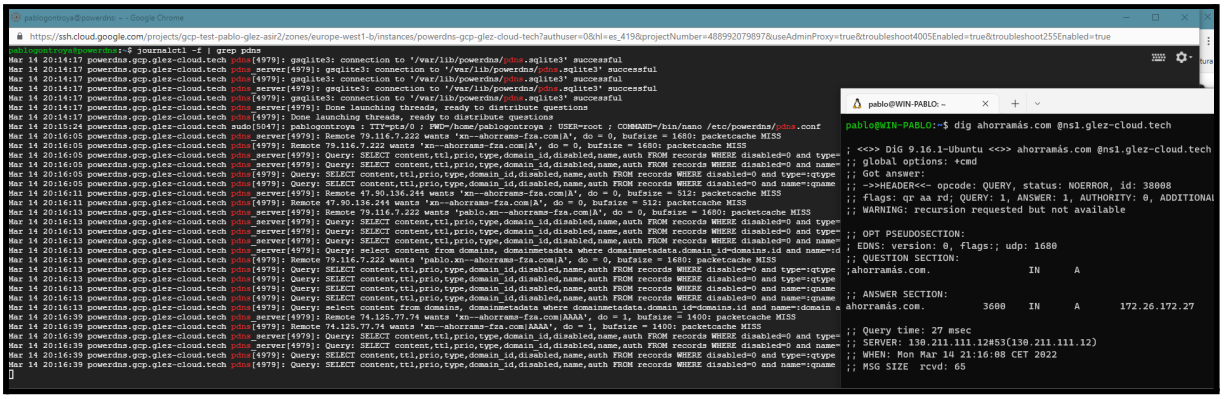
Ahora ya se mostrarán los *logs* en el diario. El problema en este momento es que los vemos mezclados con los del resto de servicios.

Una vez hecho esto, utilizando el siguiente comando podemos filtrar para ver sólo los registros de powerdns:

```
journalctl -f | grep pdns
```

[28] https://doc.powerdns.com/authoritative/settings.html#loglevel

En la imagen a continuación podemos ver una imagen de muestra con unas peticiones que hemos hecho, que justo se han mezclado con alguna de las que realizaban las pruebas de RIPE Atlas.



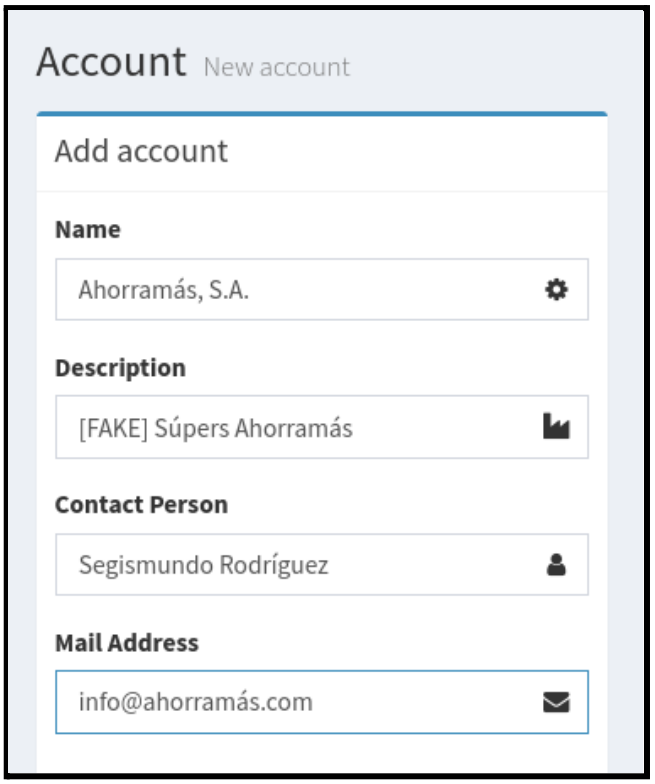
También se refleja en este documento el log generado con una consulta:

```
Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns_server[4979]: Remote 106.11.38.10 wants 'xn--ahorrams-fza.com|AAAA', do = 1, bufsize = 512: packetcache MISS

Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns[4979]: Remote 106.11.38.10 wants 'xn--ahorrams-fza.com|AAAA', do = 1, bufsize = 512: packetcache MISS

Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns_server[4979]: Query: SELECT content,ttl,prio,type,domain_id,disabled,name,auth FROM records WHERE disabled=0 and name=:qname and domain_id=:domain_id

Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns_server[4979]: Query: select content from domains, domainmetadata where domainmetadata.domain_id=domains.id and name=:domain and domainmetadata.kind=:kind

Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns[4979]: Query: SELECT content,ttl,prio,type,domain_id,disabled,name,auth FROM records WHERE disabled=0 and name=:qname and domain_id=:domain_id

Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns[4979]: Query: select content from domains, domainmetadata where domainmetadata.domain_id=domains.id and name=:domain and domainmetadata.kind=:kind

Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns_server[4979]: Remote 106.11.38.7 wants 'xn--ahorrams-fza.com|A', do = 1, bufsize = 512: packetcache MISS

Mar 14 20:27:17 powerdns.gcp.glez-cloud.tech pdns[4979]: Remote 106.11.38.7 wants 'xn--ahorrams-fza.com|A', do = 1, bufsize = 512: packetcache MISS
```

5.1.5. Dominio real, pruebas de estrés y reliability

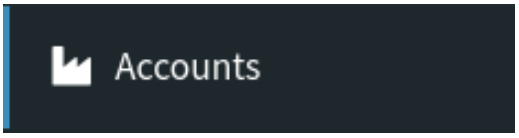
5.1.5.1. Creación de cuenta de empresa y usuario

La primera operación que vamos a realizar será crear un nuevo perfil en el panel web para que los clientes ficticios puedan gestionar sus propios registros DNS.



Existe este vídeo[29] en el que se puede ver todo el proceso de creación de usuarios. En cualquier caso, lo vamos a mostrar también en este documento, mediante capturas de pantalla.

1. En el panel lateral, hacemos clic en la pestaña *Accounts*.



2. Si es la primera vez que accedemos a esta sección de la GUI de Administración, nos la encontraremos vacía. Haremos clic en el botón *Add Account* + para crear una nueva:

3. Introducimos los datos para la nueva cuenta, tal y como podemos ver en la imagen a la izquierda de este párrafo. En este caso, nos está contratando

[29] Ver vídeo en Loom - https://www.loom.com/share/3a55cc1209434ef1955fa66692ac735e

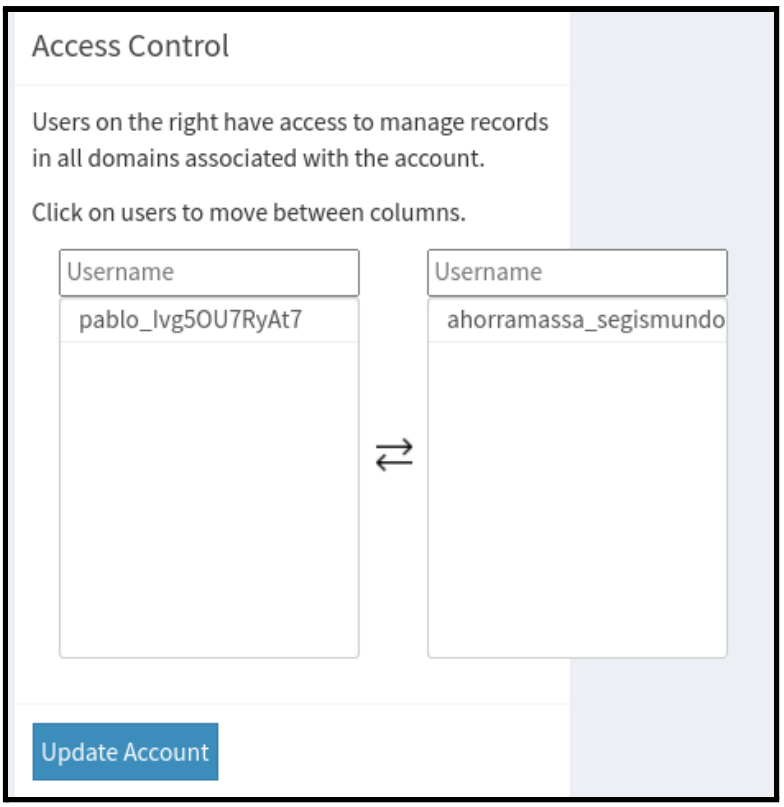
"Ahorramás, S.A." Evidentemente, es una contratación completamente falsa y solo servirá de ejemplo para este proyecto[30].

4. Después de crear la empresa, debemos crear un usuario administrador que será el usuario que pueda (además del superadministrador) gestionar las entradas DNS de las zonas.

5. Hacemos clic, dentro de las opciones del menú de la izquierda, en *users*. Introducimos el nombre y los apellidos del administrador/a de la cuenta. En nuestro caso *Segismundo Rodríguez.* En la dirección de correo electrónico, introduciremos *segismundo@ahorramás.com*. Aunque para el apartado de username no hay ninguna imposición técnica, se llega a la conclusión de que lo mejor es indicar en este a qué empresa pertenece, por tanto, se introducirá *ahorramassa_segismundo*. Para la contraseña se introducirá la siguiente generada de forma aleatoria: *5fv8riEH*ibF4dtc* . También queda reflejada en el apartado de contraseñas y accesos[31] de este documento. Hacemos clic en guardar. Ya lo tenemos disponible en la pantalla de usuarios:

6. Si bien ya tenemos creado el usuario y la empresa, no tenemos estas dos entidades vinculadas. Para hacerlo, en el menú de la izquierda navegamos hasta *Accounts* de nuevo. Aquí, pinchamos sobre el botón *edit* que disponemos en la entrada de la cuenta *ahorramassa*.
7. En la parte inferior de la pantalla, podemos ver dos tablas con usuarios. Los usuarios que desplacemos, utilizando las flechas centrales, a la derecha sí tendrán acceso a esta organización. Movemos al usuario *ahorramassa_segismundo* a esta lista:

[30] *Ahorramas, S.A* no está relacionada con el presente documento ni proyecto ni se ha visto afectada por la realización de las pruebas.
[31] Ver sección *7.3. Anexo III: Contraseñas de los servicios*.



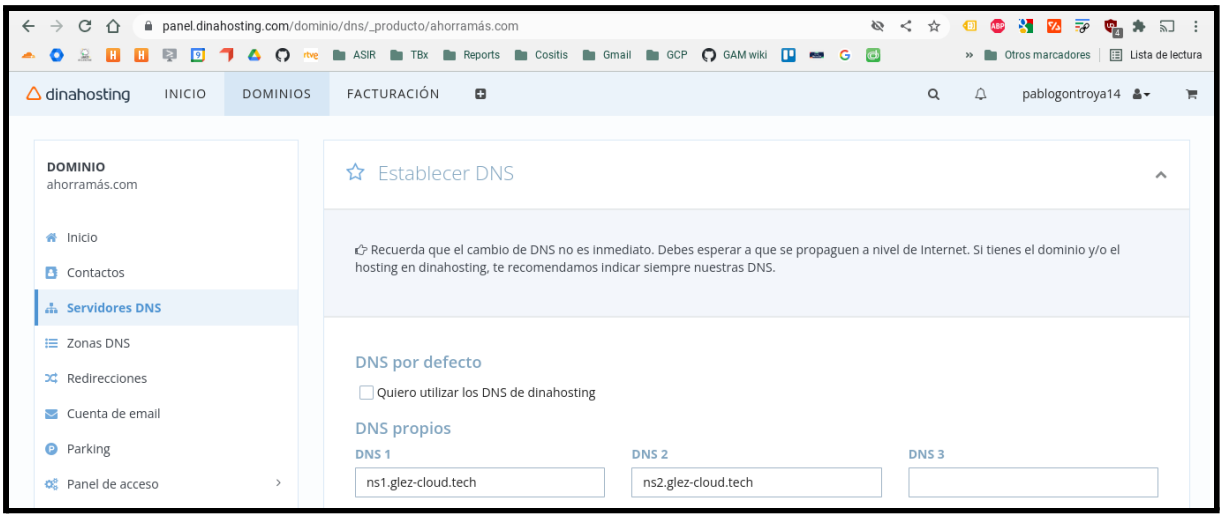
8. Hacemos clic en *Update Account* para guardar los cambios.

5.1.5.2. Dar de alta dominio y apuntado de NS. Punycode.

Como se puede ver en el apartado anterior, se trabajará como si la empresa de supermercados Ahorramás hubiera contratado los servicios de GLEZCLOUD. Evidentemente, no lo ha hecho. Para simular esto, se ha adquirido el dominio `ahorramás.com` y `ahorramás.es`. Como se puede observar, son dominios ligeramente especiales: ¡tienen tildes! Son dominios completamente diferentes a `ahorramas.com` y `ahorramas.es`, respectivamente. La creación de estos dominios es posible gracias a la implementación del estándar Punycode, desarrollado en el RFC 3492[32]. Puesto que considero que es un tema interesante, será tratado (junto con las posibilidades y riesgos que su uso conlleva) en el anexo V de este documento (*Respecto a la nomenclatura punycode*).

[32] Request for Comments (RFC) 3492 - Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA) - https://datatracker.ietf.org/doc/html/rfc3492/

Para que nuestro servidor DNS pueda resolver el dominio ahorramás.com sin que sea necesario indicar explícitamente qué servidor DNS queremos que lo resuelva, debemos editar los NS en el registrador del dominio. En nuestro caso, este registrador es DINAHOSTING. Su interfaz de gestión es sencilla, y basta con introducir los nombres que deseemos:



Por compatibilidad se han introducido dos servidores DNS en los campos habilitados para este fin. Realmente ambos resuelven a la misma dirección IP: `130.211.111.12`. Pero por compatibilidad y por seguir las recomendaciones se han introducido dos nombres de dominio diferentes.

Para ver si se han aplicado correctamente los cambios, basta con utilizar el comando whois en cualquier equipo bajo Linux, o consultar cualquier herramienta online como https://whois.domaintools.com/ y similares.

En nuestro caso, se ha utilizado el comando y el resultado ha sido el siguiente:

```
whois ahorramás.com
```

```
Domain Name: XN--AHORRAMS-FZA.COM
Registry Domain ID: 2620216271_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.dinahosting.com
Registrar URL: http://www.dinahosting.com/dominios
Updated Date: 2022-03-13T16:16:15Z
Creation Date: 2021-06-16T23:27:02Z
Registry Expiry Date: 2022-06-16T23:27:02Z
Registrar: Dinahosting s.l.
```

```
   Registrar IANA ID: 1262
   Registrar Abuse Contact Email: abuse-domains[@]dinahosting.com
   Registrar Abuse Contact Phone: +34.981040200
   Domain Status: clientDeleteProhibited
   Domain Status: clientTransferProhibited
   Name Server: NS1.GLEZ-CLOUD.TECH
   Name Server: NS2.GLEZ-CLOUD.TECH
   DNSSEC: unsigned
```

Como podemos comprobar, nuestro(s) servidor(es) DNS ya figuran como los autoritativos para la resolución de las zonas de XN--AHORRAMS-FZA.COM, que es la representación en punycode de ahorramás.com.

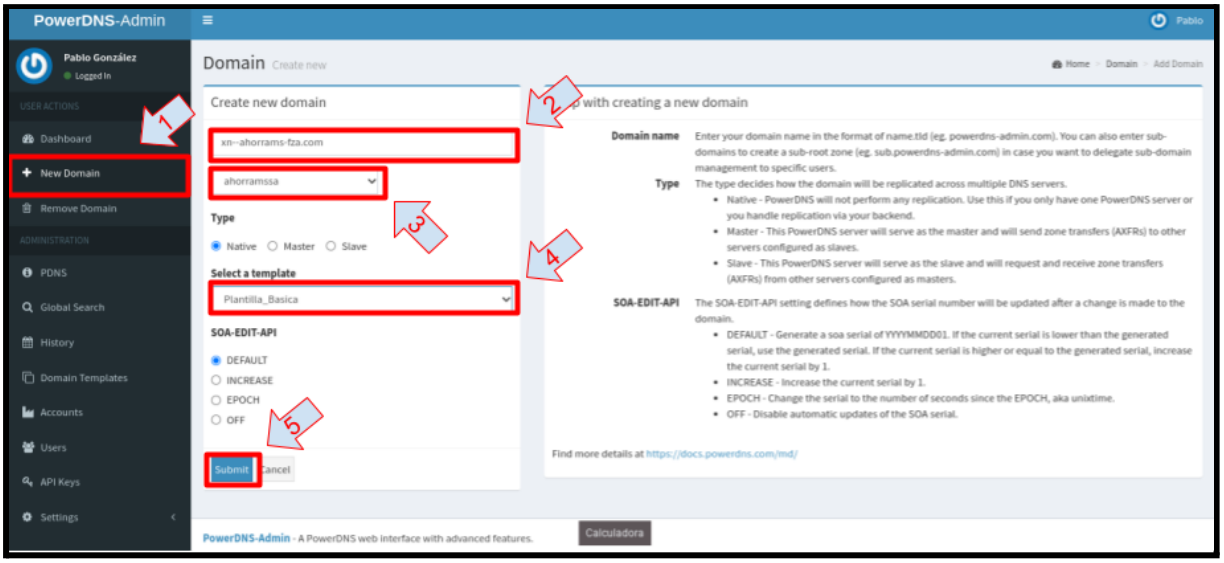
Si ahora intentamos la resolución DNS del dominio (preguntando a un servidor DNS público directamente para evitar la caché DNS que pudiera tener alguno de nuestros dispositivos locales), recibiremos un mensaje de error. Esto es debido a que en este momento ni ns1.glez-cloud.tech, ni ns2.glez-cloud.tech están configurados para resolver el dominio, por tanto rechazan la resolución. En caso de intentarlo, recibiremos un error *feo* parecido al siguiente:

```
dig ahorramás.com @1.1.1.1
```

```
; <<>> DiG 9.11.5-P4-5.1+deb10u6-Debian <<>> ahorramás.com @1.1.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 25535
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; OPT=15: 00 16 61 74 20 64 65 6c 65 67 61 74 69 6f 6e 20 78 6e 2d 2d 61
68 6f 72 72 61 6d 73 2d 66 7a 61 2e 63 6f 6d 2e ("..at delegation
xn--ahorrams-fza.com.")
; OPT=15: 00 17 31 33 30 2e 32 31 31 2e 31 31 31 2e 31 32 3a 35 33 20 72
63 6f 64 65 3d 52 45 46 55 53 45 44 20 66 6f 72 20 78 6e 2d 2d 61 68 6f
72 72 61 6d 73 2d 66 7a 61 2e 63 6f 6d 20 41 ("..130.211.111.12:53
rcode=REFUSED for xn--ahorrams-fza.com A")
;; QUESTION SECTION:
;ahorramás.com.                 IN      A

;; Query time: 34 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sun Mar 13 17:53:56 CET 2022
```

```
;; MSG SIZE  rcvd: 154
```

Para resolverlo, basta con dar de alta en nuestro servidor PowerDNS el dominio y sus respectivas entradas. Para hacerlo, hacemos clic en el menú izquierdo sobre la opción *New* Domain (Marca 1 sobre la imagen).



Acto seguido, debemos introducir el nombre de dominio que deseamos gestionar en nuestro servidor OpenDNS. Introduciremos `xn--ahorrams-fza.com`, aunque en la visualización veremos el nombre con tilde (marca 2 sobre la imagen). Como cuenta que podrá gestionar los registros de este dominio, seleccionamos *ahorramassa* de entre las opciones del desplegable (marca 3 sobre la imagen). Como plantilla a aplicar para el nuevo dominio, indicamos *Plantilla_Basica* para que automáticamente se añadan los NS, el A de prueba y los TXT de demostración.

Una vez terminado, hacemos clic en *Submit* para guardar los cambios. Si ahora volvemos a repetir la solicitud anterior, tendremos el resultado correcto:

```
dig ahorramás.com @1.1.1.1
```

```
; <<>> DiG 9.11.5-P4-5.1+deb10u6-Debian <<>> ahorramás.com @1.1.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39848
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;ahorramás.com.                 IN      A

;; ANSWER SECTION:
ahorramás.com.          3600    IN      A       172.26.172.27

;; Query time: 32 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sun Mar 13 18:07:23 CET 2022
;; MSG SIZE  rcvd: 65
```

Si lo hacemos sin forzar servidor DNS de resolución, también funciona:

```
dig ahorramás.com +short
```

```
172.26.172.27
```

Los registros TXT también funcionan correctamente. Probemos ahora con el servidor DNS público de Google:

```
dig TXT ahorramás.com +short @8.8.8.8
```

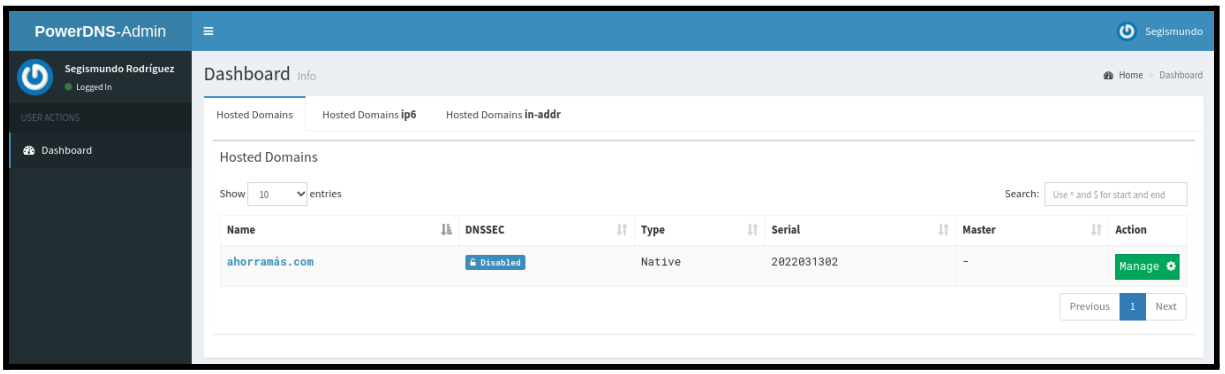
```
"Dominio funcionando en GLEZCLOUD"
"Este hosting es una maravilla"
```

Al igual con los NS:

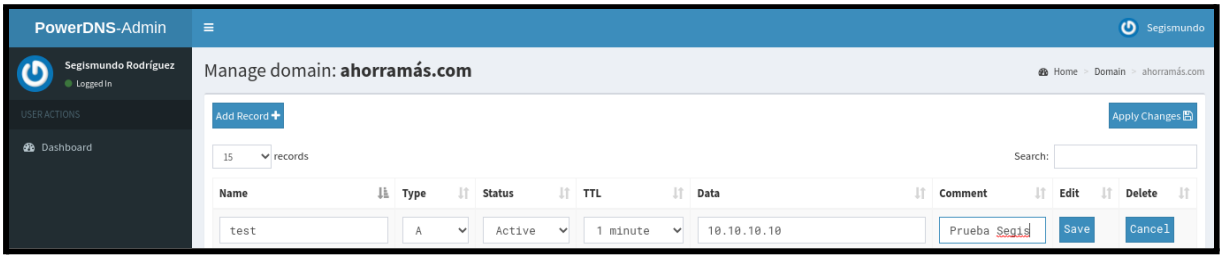
```
dig NS ahorramás.com +short @8.8.8.8
```

```
ns1.glez-cloud.tech.
ns2.glez-cloud.tech.
```

Vamos a cerrar nuestra sesión con el usuario de superadministrador, para iniciarla con *ahorramassa_segismundo*. Al hacerlo, veremos ya el dominio disponible para este usuario:



Introducimos los detalles del registro nuevo y hacemos clic en *Save*. Aplicamos los cambios utilizando el botón superior derecho.



Si ahora realizamos la consulta DNS:

```
dig test.ahorramás.com
```

```
[HEADER OMITIDA]

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;test.ahorramás.com.            IN      A

;; ANSWER SECTION:
test.ahorramás.com.     60      IN      A       10.10.10.10

;; Query time: 299 msec
;; SERVER: 100.115.92.193#53(100.115.92.193)
;; WHEN: Sun Mar 13 18:20:33 CET 2022
;; MSG SIZE  rcvd: 70
```

5.1.5.3. Pruebas de *reliability* con RIPE Atlas

RIPE Atlas es una de las plataformas de medición de parámetros de red en Internet de mayor despliegue a nivel mundial. Pone a disposición de sus miembros recursos que permiten realizar mediciones de redes. RIPE NCC es el organismo encargado de llevar adelante este proyecto[33].

La red global de sondas RIPE Atlas realiza mediciones activas acerca de la conectividad y capacidad de alcance de Internet. Esta red facilita una comprensión sin precedentes acerca del estado de Internet en tiempo real para toda la comunidad de Internet: cualquiera puede acceder a los mapas, estadísticas y resultados de las mediciones de RIPE Atlas[34].

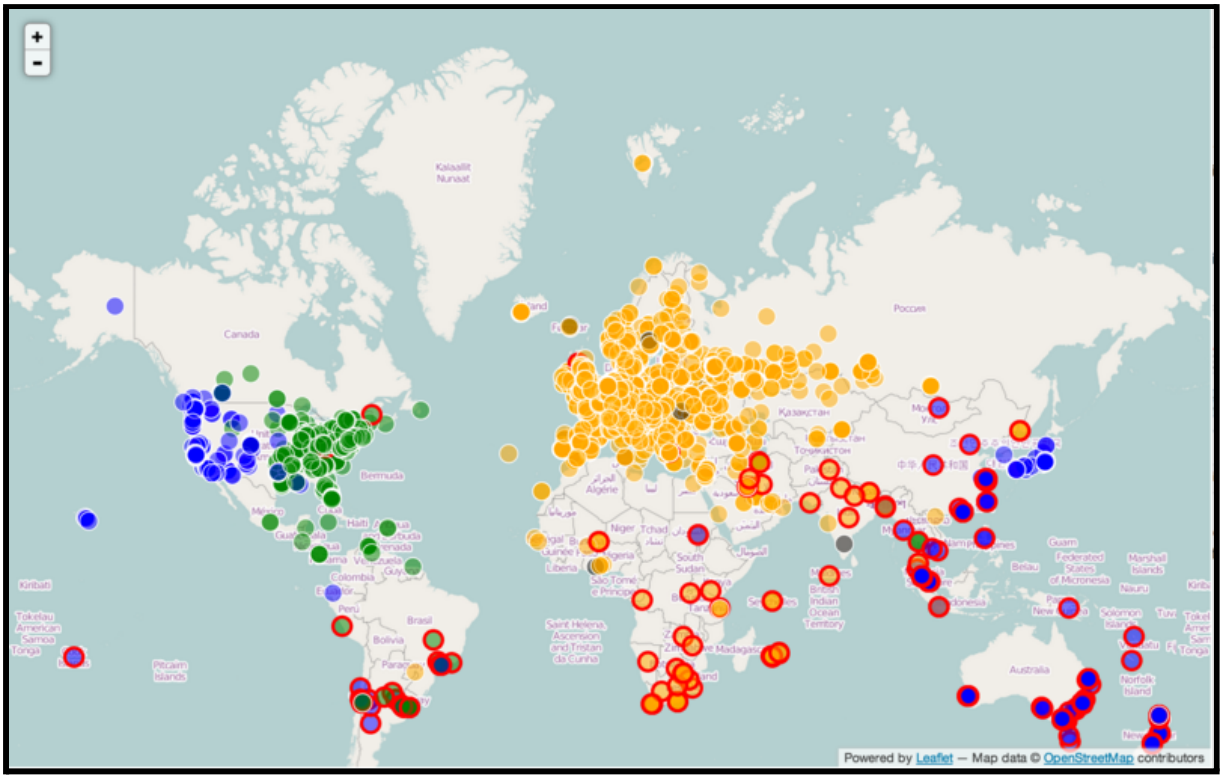
RIPE Atlas es usada por miles de personas, investigadoras/es y administradores de red para, por ejemplo[33]:

- Monitorizar continuamente la capacidad de alcance de la red desde miles de puntos de observación alrededor del mundo.
- Investigar y detectar problemas de red con controles de conectividad rápidos y flexibles.
- Crear alarmas usando los comprobadores de estado de RIPE Atlas, que pueden integrarse con sus propias herramientas de monitorización.
- Medir la capacidad de respuesta de su infraestructura DNS o la de los servidores raíz.
- Probar la conectividad IPv6

Como ejemplos, podemos destacar el estudio que hizo la fundación Wikimedia (que gestiona, entre otros proyectos, Wikipedia) para saber dónde posicionar puntos de presencia (PoPs) para su Red de Distribución de Contenido (CDN) internacional[35] y desde cuál servir a los diferentes usuarios[36].

[33] https://www.lacnic.net/1000/1/lacnic/ripe-atlas-en-latinoamerica-y-caribe
[34] https://www-static.ripe.net/static/rnd-ui/atlas/media/brochures/RIPE-Atlas-probes-2015_Spanish.pdf
[35] https://diff.wikimedia.org/2014/07/09/how-ripe-atlas-helped-wikipedia-users/
[36] https://phabricator.wikimedia.org/diffusion/ODNS/browse/master/config-geo;43be064b6e12e8bf67c8ccfac46749a0ba193d19

Además de para generar un mapa (real) con las latencias para sus distintos servidores como se puede ver en la imagen a continuación, pudieron monitorizar sus sistemas y sistemas para conocer la accesibilidad que realmente tenían desde los distintos *probes de* RIPE ATLAS que había desplegados globalmente[37].



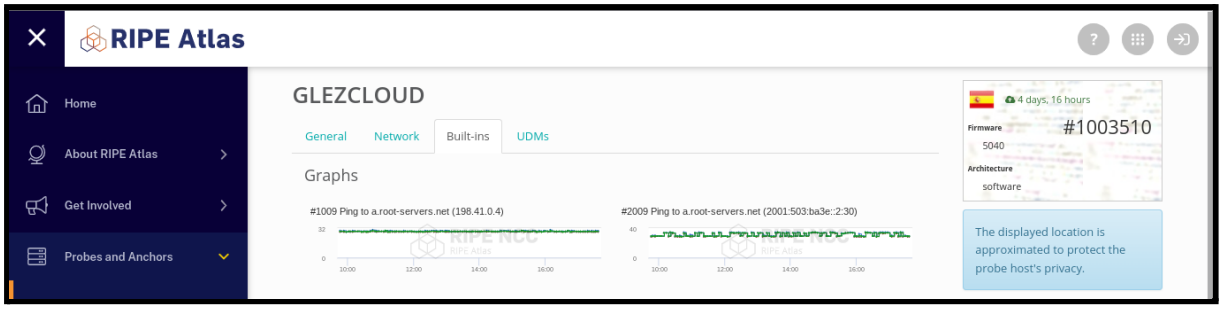
También hay otros estudios muy interesantes en los que podemos ver que para dos dispositivos de red localizados dentro del territorio nacional de un mismo país, el tráfico viaja fuera de sus fronteras y por varios IXP (Internet eXchange Points)[38].

Recientemente también se han llevado a cabo estudios sobre la resiliencia y disponibilidad del acceso a Internet en Ucrania, en relación al ataque por parte de la Federación de Rusia. Como ejemplos encontramos los siguientes:

- The Resilience of the Internet in Ukraine[39]
- The Ukrainian Internet[40]

[37] https://labs.ripe.net/author/suzanne_taylor_muzzin/introducing-ripe-atlas-status-checks/
[38] https://labs.ripe.net/author/emileaben/measuring-countries-and-ixps-with-ripe-atlas/
[39] https://labs.ripe.net/author/emileaben/the-resilience-of-the-internet-in-ukraine/
[40] https://labs.ripe.net/author/emileaben/the-ukrainian-internet/

El sistema funciona en base a créditos. Por alojar un *probe*, asegurar su disponibilidad y en función de los resultados que entrega el operador recibe unos créditos. Aquí[41] se puede comprobar el estado del *probe* que estoy alojando:

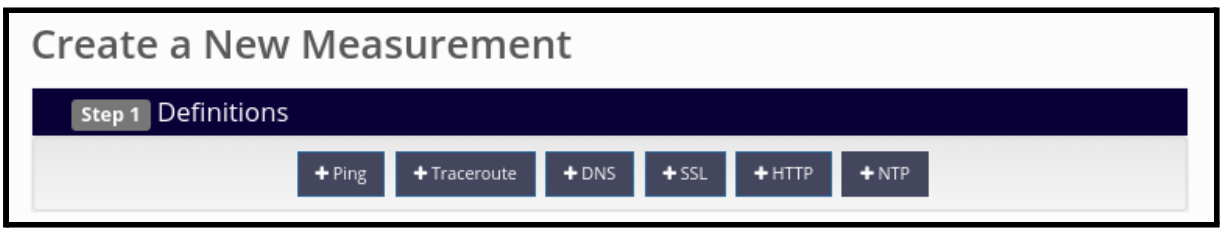


Como regalo para el lector que haya llegado hasta aquí leyendo, puede utilizar este código para recibir un millón de créditos para probar RIPE Atlas: *DEPLOYATHON2021*. Si este dejara de funcionar, no dudes en escribirme un correo electrónico a pfc-altas-credits [@] gonzaleztroyano.es con tu ID de cuenta y el número de créditos que necesitas para probarlo. ¡Estaré encantado de enviarte unas decenas de miles de créditos para que puedas empezar!

En nuestro caso, lo que haremos será comprobar la correcta resolución de nuestro servidor DNS desde 10 ubicaciones en el mundo. De esta forma veremos a lo largo del tiempo si ha sido posible la correcta resolución del dominio. También realizaremos una prueba de estrés con muchas más solicitudes de resolución simultáneas.

Para la *measurement*, prueba continuada seguiremos el siguiente proceso:

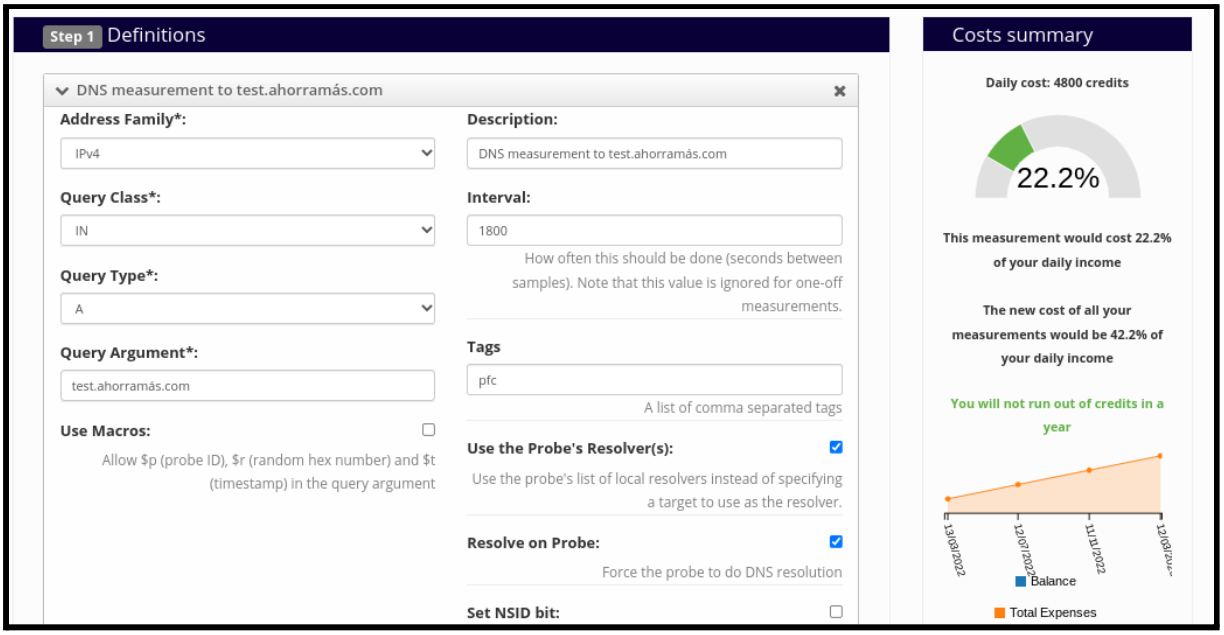
1. En tipo de prueba, indicamos DNS:



2. Una vez definido el tipo de prueba, vamos con los detalles de esta:

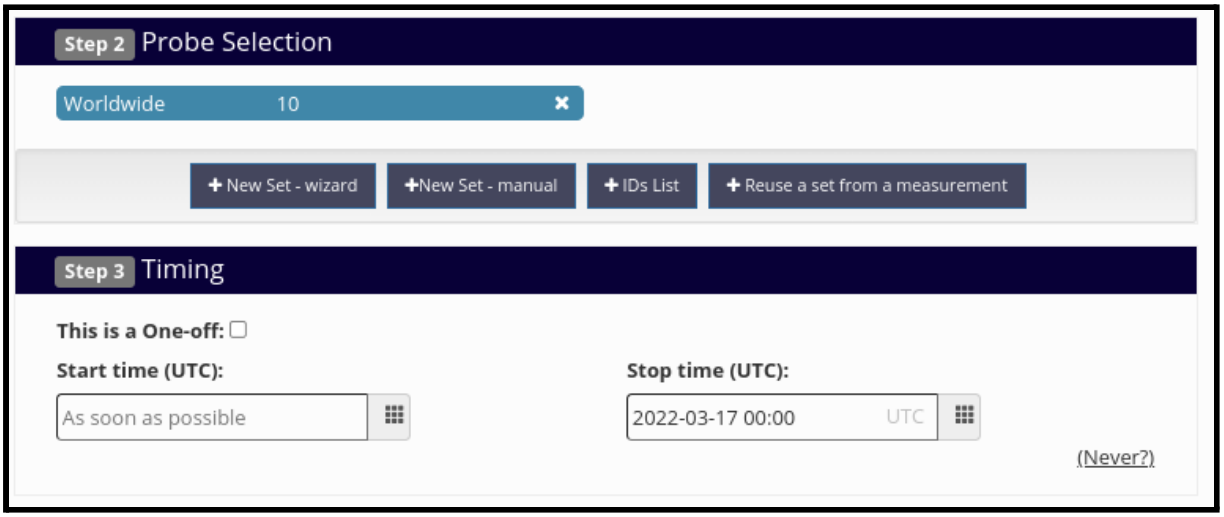
[41] https://atlas.ripe.net/probes/1003510/#tab-general

a. Será una consulta en IPv4, pues el servidor no tiene acceso a IPv6.
b. Se realizará la consulta cada 1800 segundos (30 minutos).
c. Será una consulta al registro A del dominio test.ahorramás.com.
d. Para tenerlas localizadas, aplicaremos la etiqueta “pfc”.
e. Forzaremos la resolución DNS en el propio *probe*.



3. Por último, indicamos cuántas *probes* y en qué región serán las encargadas de preguntar de forma continuada al servidor DNS.

4. También marcaremos la temporalidad de la medición. En nuestro caso, la iniciaremos al momento y la estaremos probando durante aproximadamente 5 días.



Los resultados están disponibles en este enlace[42]. A continuación los vamos a estudiar.

Los últimos resultados de latencia proporcionados por las pruebas se pueden ver en la siguiente imagen:

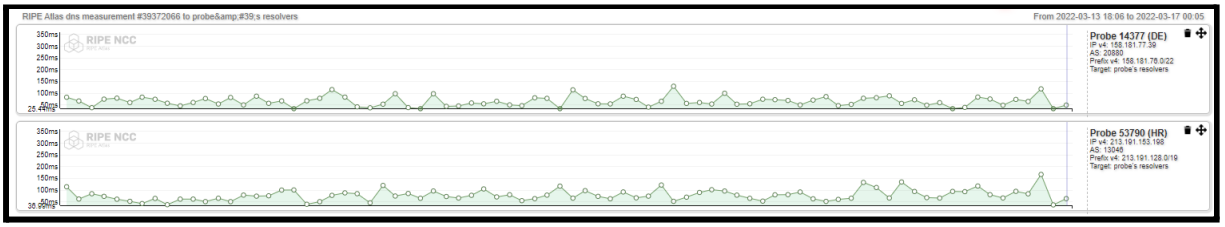
Probe	ASN (IPv4)	ASN (IPv6)		Time (UTC)	Answer	Response Time
1002814	206238			2022-03-16 23:36	NOERROR	19.255
1002309	204754			2022-03-16 23:36	NOERROR	26.994
14377	20880			2022-03-16 23:36	NOERROR	27.96
53790	13046			2022-03-16 23:37	NOERROR	40.323
1003161	7029			2022-03-16 23:36	NOERROR	112.798
1000473	21859			2022-03-16 23:36	NOERROR	144.806
51335	35891			2022-03-16 23:36	REFUSED	201.827
1002669	17117			2022-03-16 23:36	NOERROR	284.768
52219	19009			2022-03-16 23:36	NOERROR	373.261
30219	9790	9790		2022-03-16 23:36	NOERROR	424.561

Tal y como se puede observar, todas las pruebas han ofrecido resultados satisfactorios. Las 10 peticiones de resolución DNS han recibido respuesta "NOERROR". Es decir, el servidor ha entendido la solicitud y ha ofrecido respuesta.

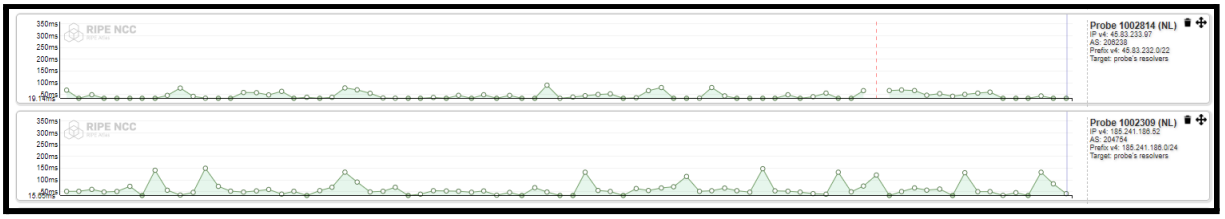
La menor latencia ha sido reflejada por la prueba #1002814[43]. Está situada en Países Bajos y la latencia no llega a los 20 ms. La media de la latencia en las pruebas de europa ha sido de unos 26 ms, una métrica muy buena.

En comparación, la latencia más alta la encontramos reflejada en la *probe* #30219[44], situada físicamente Nueva Zelanda. Aún así la latencia ha sido de unos 400 ms, nada mal.

Como vemos para las pruebas europeas, los resultados han sido muy estables:

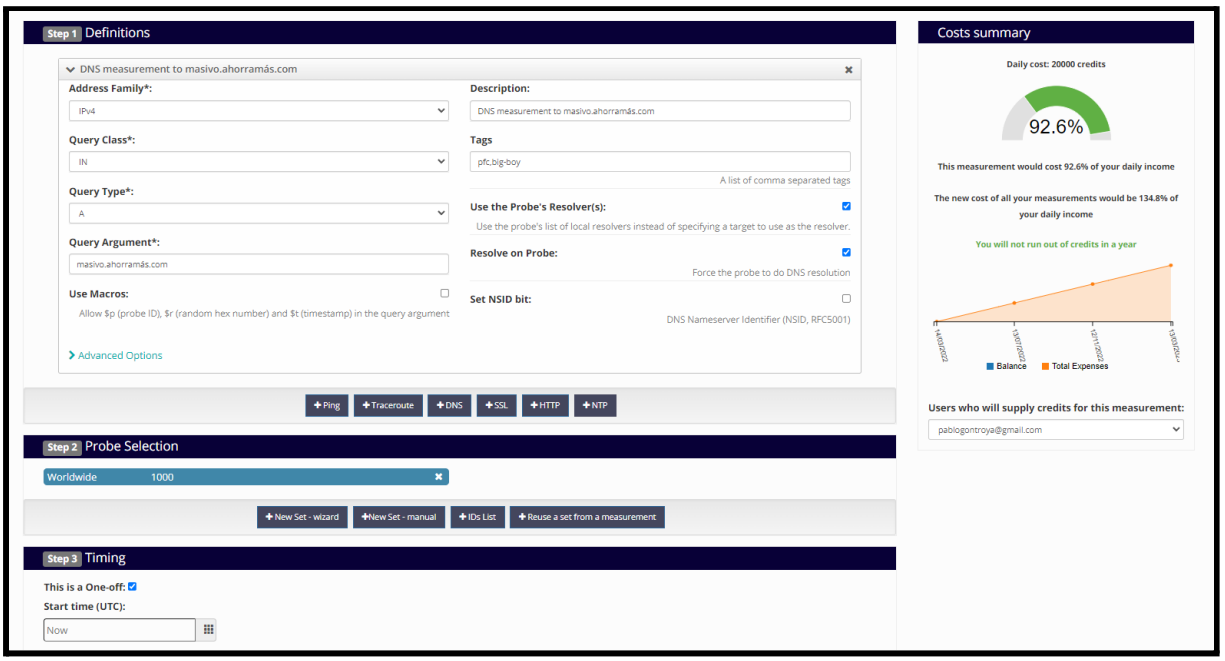


[42] https://atlas.ripe.net/measurements/39372066/#general
[43] https://atlas.ripe.net/frames/probes/1002814/
[44] https://atlas.ripe.net/frames/probes/30219/



Además de las pruebas de *reliability*, como se ha comentado anteriormente, se va a realizar una prueba de estrés.

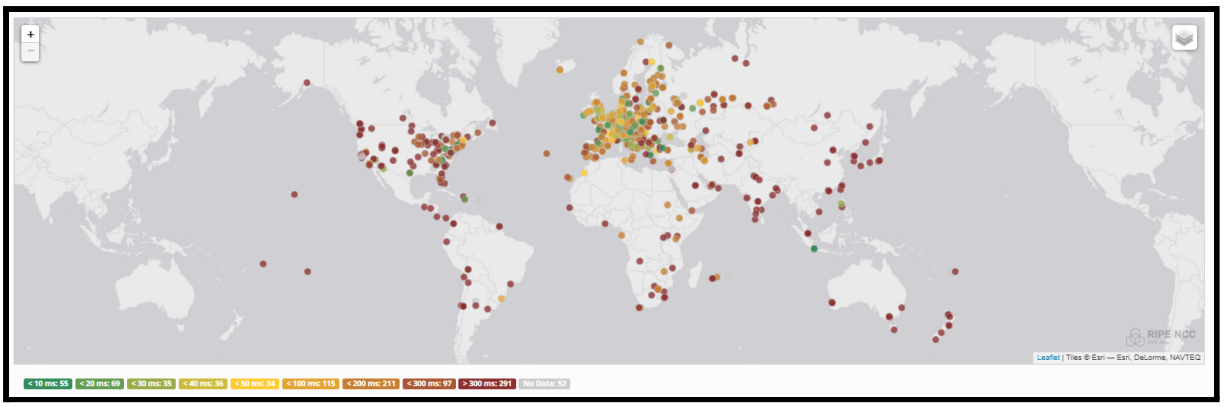
En esta prueba, 1000 máquinas solicitarán a la vez la resolución A de la entrada masivo.ahorramás.com. Esta es la configuración en la interfaz de RIPE Atlas:



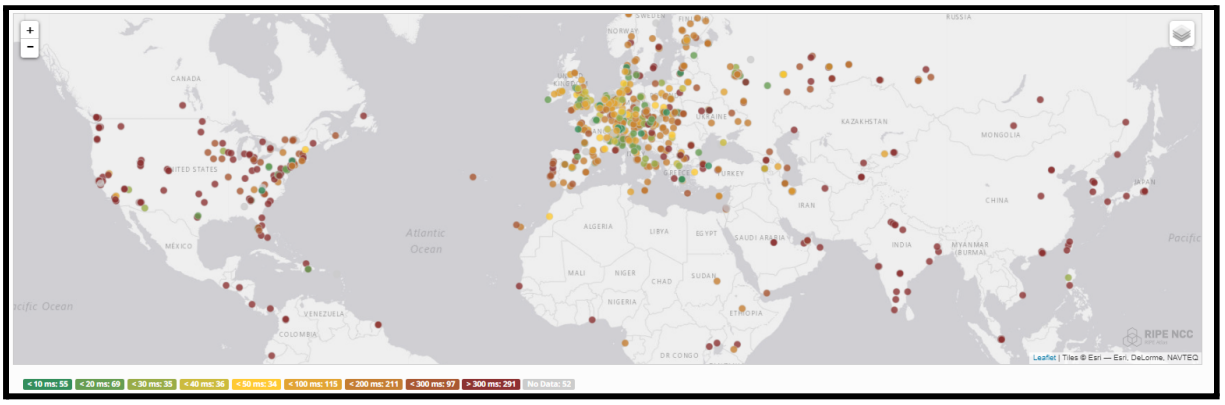
Como se puede observar en la imagen, se solicita que 1000 *probes* participen. Se lanza la solicitud y se espera para ver los resultados. La medición tiene el identificador número 39397876 y se puede consultar desde este enlace[45].

Únicamente han participado 996 *probes*, geográficamente muy distribuidas por el mundo. El servidor en Google Cloud Platform está ubicado físicamente en Bruselas, y observando las latencias está claro que la regla general confirma la ubicación de estas. Veamos el mapa:

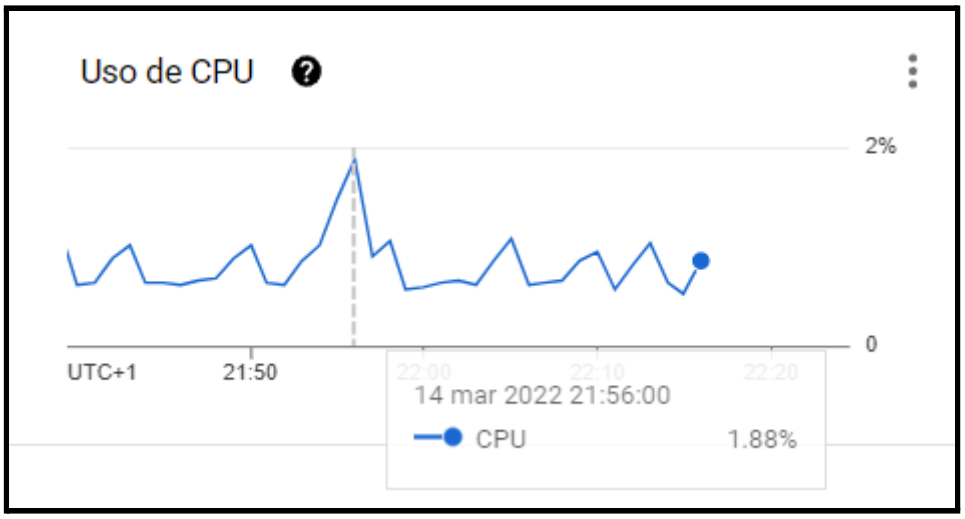
[45] https://atlas.ripe.net/frames/measurements/39397876/



Obviando ahora las islas aisladas y centrándonos únicamente en las regiones de EMEA[46] y US:



Respecto a cómo se han reflejado estas 1000 peticiones en el uso de la instancia, podemos ver que prácticamente no se ha visto afectada:



[46] *Europe, Middle East and Asia*.

Hemos descargado los registros, que están disponibles en el repositorio de Github[47], después hemos utilizado la CLI de ipinfo.io[48] para tratar los datos. Habiéndolos guardado en el archivo “ips”, se ejecuta el siguiente comando:

```
pablo@WIN-PABLO:~$ cat ips | ipinfo grepip -o -x
185.191.34.215
185.191.34.215
185.191.34.215
```

Después, ejecutamos el siguiente comando para ver un resumen de los datos de las IPs contenidas en el archivo:

```
cat ips | ipinfo grepip -o -x | ipinfo summarize
```

Esto es lo que nos devuelve el comando:

```
Summary
- Total   1944
- Unique  760
- Anycast 16
- Bogon   0
- Mobile  14
- VPN     34
- Proxy   0
- Hosting 672
- Tor     0
- Relay   290

Top ASNs
- AS13335 Cloudflare, Inc.    312 (16.0%)
- AS15169 Google LLC          190 (9.8%)
- AS42 WoodyNet               108 (5.6%)
- AS36692 Cisco OpenDNS, LLC 68 (3.5%)
- AS16509 Amazon.com, Inc.    60 (3.1%)

Top Usage Types
- ISP       792 (40.7%)
- Hosting   672 (34.6%)
- Business  258 (13.3%)
- Education 60 (3.1%)

Top Routes
- 172.253.0.0/16 (AS15169)  78 (4.0%)
- 185.98.114.0/24 (AS43754) 42 (2.2%)
- 172.217.0.0/16 (AS15169)  40 (2.1%)
- 74.125.176.0/20 (AS15169) 36 (1.9%)
- 166.111.8.0/24 (AS4538)   30 (1.5%)
```

```
Top Cities
- Frankfurt am Main, Hesse, DE 64 (3.3%)
- Tehran, Tehran, IR           58 (3.0%)
- Paris, Île-de-France, FR     52 (2.7%)
- Moscow, Moscow, RU           50 (2.6%)
- Singapore, Singapore, SG     48 (2.5%)

Top Regions
- Hesse, DE          70 (3.6%)
- England, GB        66 (3.4%)
- Tehran, IR         58 (3.0%)
- Île-de-France, FR  54 (2.8%)
- Moscow, RU         50 (2.6%)

Top Carriers
- NOS      8 (0.4%)
- du       2 (0.1%)
- Bouygues 2 (0.1%)
- Globe    2 (0.1%)

Top Domains
- pch.net     78 (4.0%)
- opendns.com 42 (2.2%)
- asiatech.ir 42 (2.2%)
- strln.net   32 (1.6%)

Top Countries
- United States 368 (18.9%)
- Russia        146 (7.5%)
- Germany       132 (6.8%)
- France        76 (3.9%)
- Iran          74 (3.8%)
```

[47] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/1-dns/downloaded-logs-20220314-222634.csv
[48] https://github.com/ipinfo/cli

5.1.6. Configuración de servidor secundario

5.1.6.1. Creación de la instancia

Se crea la instancia `powerdns-2-gcp-glez-cloud-tech`. El FQDN de la misma será `powerdns-2.gcp.glez-cloud.tech`. Para asegurar la disponibilidad del servicio, se establecerá en la región de Londres (la primera máquina se encuentra físicamente en Bélgica). En el hipotético caso de que la región de Bélgica caiga, el servicio se mantendría ininterrumpido.

El dominio público con el que se consultará será el servidor DNS secundario publicado en el whois: `ns2.glez-cloud.tech`.

Creamos una instancia, esta vez con del tipo e2-medium (también tiene 2 vCPU pero 4GB de RAM en vez de 8 GB). Esto es posible gracias a que esta máquina no tendrá docker, portainer ni la interfaz GUI web de gestión. Por tanto, los requisitos son mucho más limitados.

Por tanto, no permitiremos el tráfico HTTP, ni HTTPS. Eliminamos, pues no son necesarias para esta VM, las etiquetas de red `portainer` y `opendns-admin`.

También podemos utilizar el siguiente comando de `gcloud` para hacerlo de forma automática:

```
gcloud compute instances create powerdns-2-gcp-glez-cloud-tech
--project=gcp-test-pablo-glez-asir2 --zone=europe-west2-c --machine-type=e2-medium
--network-interface=network-tier=PREMIUM,subnet=default --maintenance-policy=MIGRATE
--service-account=488992079897-compute@developer.gserviceaccount.com
--scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com
/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googlea
pis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,
https://www.googleapis.com/auth/trace.append --tags=powerdns-public
--create-disk=auto-delete=yes,boot=yes,device-name=powerdns-gcp-glez-cloud-tech,image=pr
ojects/ubuntu-os-cloud/global/images/ubuntu-1804-bionic-v20220302,mode=rw,size=20,type=p
rojects/gcp-test-pablo-glez-asir2/zones/europe-west1-b/diskTypes/pd-balanced
--no-shielded-secure-boot --shielded-vtpm --shielded-integrity-monitoring
--reservation-affinity=any
```

Si ahora solicitamos la resolución de los NS del dominio ahorramás.com vemos:

```
pablo@WIN-PABLO:~$ dig NS ahorramás.com +short
ns2.glez-cloud.tech.
ns1.glez-cloud.tech.
```

Ahora para los NS que nos ha devuelto la consulta anterior:

```
pablo@WIN-PABLO:~$ dig A ns{1..2}.glez-cloud.tech. +short
130.211.111.12
34.89.47.92
```

5.1.6.2. Configuración del servidor

Nos conectamos por SSH al servidor y realizaremos pasos similares a los que hemos realizado para la configuración del servidor principal en el apartado 5.1.2. Instalación de PowerDNS de este documento.

Antes de nada, la instalación de los paquetes:

```
sudo apt update -y && sudo apt upgrade -y
sudo apt install pdns-server pdns-backend-sqlite3 -y
```

Configuración de PowerDNS:

```
echo -e "#######\napi=yes\napi-key=JmnWB4iiphR6FyzygJ3sdrx1u5OCas\n
api-logfile=/var/log/pdns.log\n#######\nwebserver=yes\n
webserver-address=0.0.0.0\nwebserver-allow-from=0.0.0.0/0,127.0.0.1\n
webserver-port=8081\n#######\n
setgid=pdns\nsetuid=pdns\n#######\nlaunch=gsqlite3\n
gsqlite3-database=/var/lib/powerdns/pdns.sqlite3\n\n#######\n
loglevel=6\nquery-logging=yes\nlog-dns-details=yes\n
log-dns-queries=yes\nlogging-facility=0\ndisable-syslog=no" >
/etc/powerdns/pdns.conf
```

Creamos la Base de datos:

```
sudo mkdir /var/lib/powerdns
wget https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-PFC/main/1-dns/schema.sqlite3.sql
sudo sqlite3 /var/lib/powerdns/pdns.sqlite3 < schema.sqlite3.sql
sudo chown -R pdns:pdns /var/lib/powerdns
```

Liberamos el puerto 53/UDP:

```
sudo systemctl disable systemd-resolved
sudo systemctl stop systemd-resolved
```

Eliminamos el enlace simbólico (si lo tuviera) del archivo `/etc/resolv.conf` y renombramos el archivo por si hubiera que recuperarlo. Creamos el nuevo archivo.

```
ls -lh /etc/resolv.conf  # Para ver si existiera el enlace
sudo mv /etc/resolv.conf /etc/resolv.conf.bak
echo "nameserver 1.1.1.1" | sudo tee /etc/resolv.conf
echo "127.0.0.1 powerdns-2.gcp.glez-cloud.tech." >> /etc/hosts
echo "127.0.0.1 powerdns-2" >> /etc/hosts
echo "powerdns-2.gcp.glez-cloud.tech" > /etc/hostname
```

Reiniciemos el servicio y habilitémosle para reiniciarse:

```
sudo systemctl restart pdns
sudo systemctl enable pdns
```

Comprobamos que está funcionando correctamente el servicio:

```
root@powerdns-2:~# sudo systemctl status pdns
● pdns.service - PowerDNS Authoritative Server
   Loaded: loaded (/lib/systemd/system/pdns.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-03-14 22:25:56 UTC; 4s ago
     Docs: man:pdns_server(1)
           man:pdns_control(1)
           https://doc.powerdns.com
 Main PID: 21104 (pdns_server)
    Tasks: 10 (limit: 4662)
   CGroup: /system.slice/pdns.service
           └─21104 /usr/sbin/pdns_server --guardian=no --daemon=no --disable-syslog --log-timestamp=no --write-pid=

Mar 14 22:25:56 powerdns-2 pdns_server[21104]: Set effective user id to 112
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: Could not retrieve security status update for '4.1.1-1.Ubuntu' on 'a
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: Creating backend connection for TCP
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: gsqlite3: connection to '/var/lib/powerdns/pdns.sqlite3' successful
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: About to create 3 backend threads for UDP
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: gsqlite3: connection to '/var/lib/powerdns/pdns.sqlite3' successful
Mar 14 22:25:56 powerdns-2 systemd[1]: Started PowerDNS Authoritative Server.
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: gsqlite3: connection to '/var/lib/powerdns/pdns.sqlite3' successful
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: gsqlite3: connection to '/var/lib/powerdns/pdns.sqlite3' successful
Mar 14 22:25:56 powerdns-2 pdns_server[21104]: Done launching threads, ready to distribute questions
```

5.1.6.3. Configuración de la delegación y pruebas

En el servidor principal, editamos el archivo de configuración de PowerDNS[49] [50] añadiendo lo siguiente:

```
master=yes
also-notify=34.89.47.92
```

[49] https://doc.powerdns.com/authoritative/settings.html#setting-secondary
[50] https://doc.powerdns.com/authoritative/modes-of-operation.html#secondary-operation

En el servidor secundario, editamos el archivo de configuración de PowerDNS:

```
slave=yes
allow-unsigned-notify=yes
```

Ejecutamos el siguiente comando:

```
pdnsutil set-meta xn--ahorrams-fza.com. ALSO-NOTIFY ns2.glez-cloud.tech.
```

```
root@powerdns:~# pdnsutil set-meta xn--ahorrams-fza.com. ALSO-NOTIFY ns2.glez-cloud.tech.
Mar 14 22:42:07 Reading random entropy from '/dev/urandom'
Mar 14 22:42:07 gsqlite3: connection to '/var/lib/powerdns/pdns.sqlite3' successful
Mar 14 22:42:07 gsqlite3: connection to '/var/lib/powerdns/pdns.sqlite3' successful
Mar 14 22:42:07 Query: select id,name,master,last_check,notified_serial,type,account from domains where name=:domain
Mar 14 22:42:07 Query: SELECT content,ttl,prio,type,domain_id,disabled,name,auth FROM records WHERE disabled=0 and type=:qtype and name=:qname
Mar 14 22:42:07 Query: delete from domainmetadata where domain_id=(select id from domains where name=:domain) and domainmetadata.kind=:kind
Mar 14 22:42:07 Query: insert into domainmetadata (domain_id, kind, content) select id, :kind, :content from domains where name=:domain
Set 'xn--ahorrams-fza.com' meta ALSO-NOTIFY = ns2.glez-cloud.tech.
```

Y por si fuera necesario, reiniciamos servicio y relanzamos la actualización:

```
service pdns reload
pdns_control notify-host xn--ahorrams-fza.com. ns2.glez-cloud.tech.
```

Realizamos la consulta directamente al servidor secundario:

```
; <<>> DiG 9.16.1-Ubuntu <<>> A ahorramás.com @ns2.glez-cloud.tech
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14092
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1680
;; QUESTION SECTION:
;ahorramás.com.                 IN      A

;; ANSWER SECTION:
ahorramás.com.          3600    IN      A       172.26.172.27

;; Query time: 32 msec
;; SERVER: 34.89.47.92#53(34.89.47.92)
;; WHEN: Tue Mar 15 00:07:17 CET 2022
;; MSG SIZE  rcvd: 65
```

5.1.7. Securización DNS Admin GUI utilizando Cloudflare

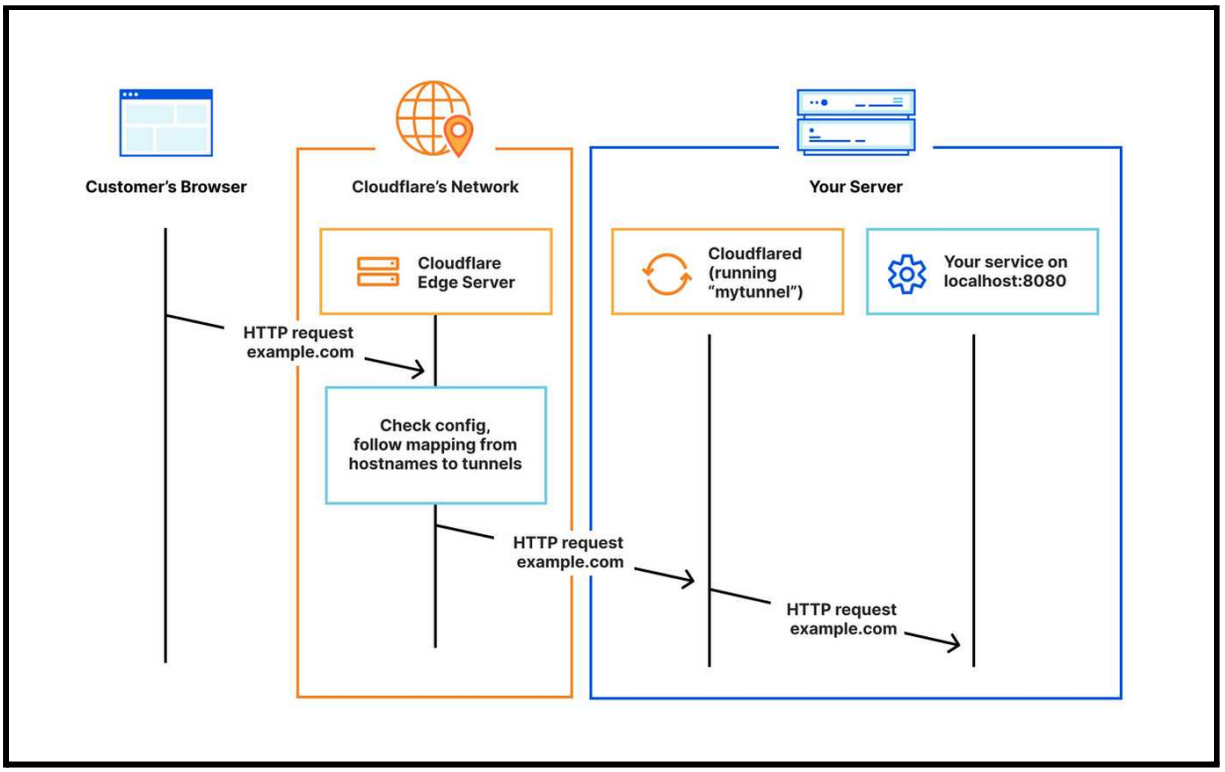
5.1.7.1. Qué es *Cloudflare Zero Trust Network Access*

Para este apartado del proyecto se utilizará el servicio de Cloudflare Tunnel (renombrado de forma reciente a *Cloudflare Zero Trust Network Access*[51]) para securizar la interfaz web de gestión.

Utilizar Cloudflare tiene una serie de ventajas:

- Mitigación DoS y DDoS. Gracias a la funcionalidad de Proxy inverso.
- Encriptación y certificado HTTPS de forma automática.
- No ser bloqueados por el firewall del instituto.
- Posibilidad de aplicar reglas personalizadas en el borde de la red.
- Gestión DNS de forma sencilla y programática gracias a las APIs públicas[52].

Para entender cómo funciona es muy útil el siguiente gráfico:



[51] https://www.cloudflare.com/es-es/products/zero-trust/zero-trust-network-access/
[52] Bash Script to dynamically update a DNS record (DDNS) using Cloudflare API https://gist.github.com/gonzaleztroyano/d86210915347f1c9ec1ceb940a5ade0c

5.1.7.2. Instalar y autorizar el agente *cloudflared*

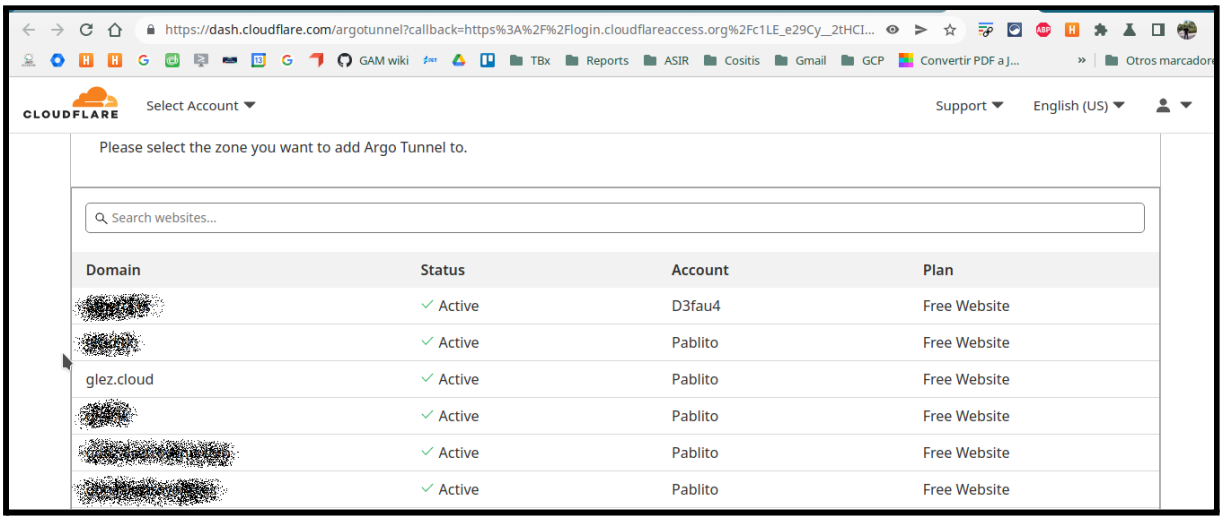
En el lado del servidor debemos instalar el agente *cloudflared*, que podemos descargar desde GitHub[53]. Ejecutaremos los siguientes comandos, con permisos de superusuario:

```
wget https://github.com/cloudflare/cloudflared/releases/latest/download/cloudflared-linux-amd64.deb
sudo dpkg -i cloudflared-linux-amd64.deb
```

Una vez instalado, debemos autorizar el agente para que sepa con qué cuenta de Cloudflare estamos a punto de configurarlo:

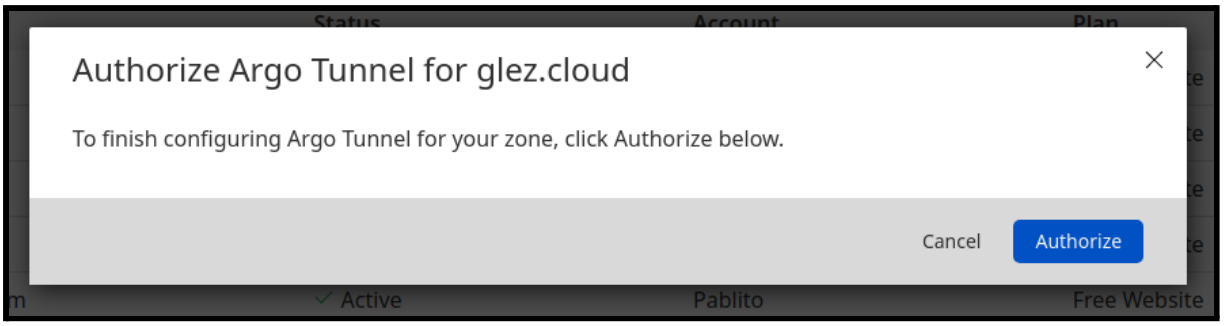
```
cloudflared tunnel login
```

Nos aparecerá en la terminal una dirección (URL) que deberemos abrir en un navegador web para autorizar el agente. Si no tuviéramos iniciada la sesión en nuestra cuenta de Cloudflare, deberemos iniciarla. Una vez hecho esto, veremos una lista con nuestros sitios web (dominios realmente) conectados. Debemos seleccionar, de entre todos los que vemos, el dominio que queremos utilizar para esta conexión:



[53] https://github.com/cloudflare/cloudflared

Por seguridad, Cloudflare nos solicitará que reconfirmemos el dominio que queremos conectar. Hacemos clic en *Authorize:*



El certificado necesario para configurar los túneles y comunicarse con Cloudflare es guardado en esta ruta: `/home/pablogontroya/.cloudflared/cert.pem`

5.1.7.3. Creación del túnel

Utilizamos el siguiente comando para crear el túnel:

```
cloudflared tunnel create dns-admin
```

Comprobamos que se ha creado correctamente:

```
cloudflared tunnel list
```

El resultado será similar al siguiente:

```
ID                                   NAME CREATED              CONNECTIONS
7697edc3-64e4-4ffd-9e6c-3bf4fa79abed dns 2022-03-17T00:12:37Z 2xFRA, 2xMAD
```

Como vemos, se han creado 4 conexiones en total. Dos de ellas a un centro de datos de Cloudflare en Madrid y otras dos a otro centro de datos de Cloudflare en Frankfurt. Alta resiliencia, por si fallase uno de los CPDs de Cloudflare (poco probable).

Crearemos el archivo `~.cloudflared/config.yml` con el siguiente contenido:

```
tunnel: 7697edc3-64e4-4ffd-9e6c-3bf4fa79abed
credentials-file: /home/pablogontroya/.cloudflared/7697edc3-64e4-4ffd-9e6c-3bf4fa79abed.json

ingress:
  - hostname: ssh-ns1.glez.cloud
    service: ssh://localhost:22
  - hostname: portainer-ns1.glez.cloud
    service: http://localhost:9000
  - hostname: manage-dns.glez.cloud
    service: http://localhost:80
  - hostname: manage-dns-info.glez.cloud
    service: http://localhost:8081
  - service: http_status:404
```

Validamos el archivo:

```
cloudflared tunnel ingress validate
```

Lanzamos el túnel para realizar pruebas:

```
cloudflared tunnel run
```

5.1.7.4. Creación de registros y activación como servicio

Debemos crear registros CNAME que apunten a [TunnelID].cfargotunnel.com bajo los subdominios seleccionamos.

```
manage-dns.glez.cloud.              IN      CNAME   7697edc3-64e4-4ffd-9e6c-3bf4fa79abed.cfargotunnel.com.
manage-portainer.glez.cloud.        IN      CNAME   7697edc3-64e4-4ffd-9e6c-3bf4fa79abed.cfargotunnel.com.
manage-dns-info.glez.cloud.         IN      CNAME   7697edc3-64e4-4ffd-9e6c-3bf4fa79abed.cfargotunnel.com.
```

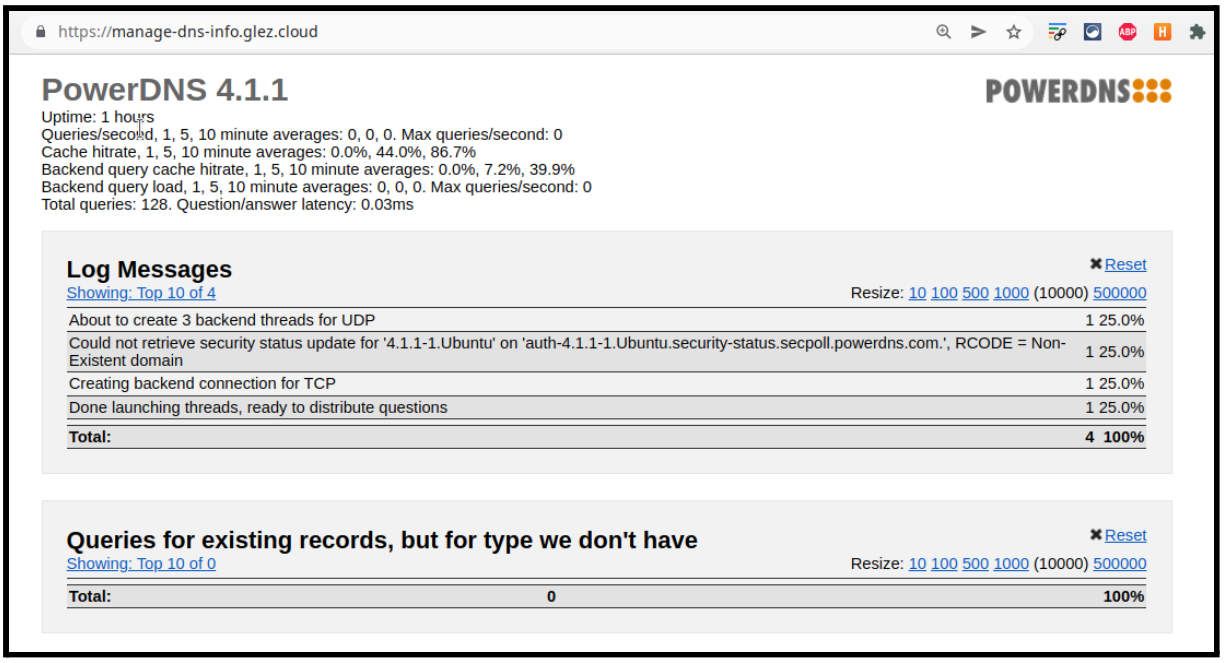
Activaremos el agente de cloudflare como servicio para asegurarnos de que se mantiene activo durante los reinicios:

```
sudo cloudflared service install
sudo systemctl start cloudflared
sudo systemctl enable cloudflared
```

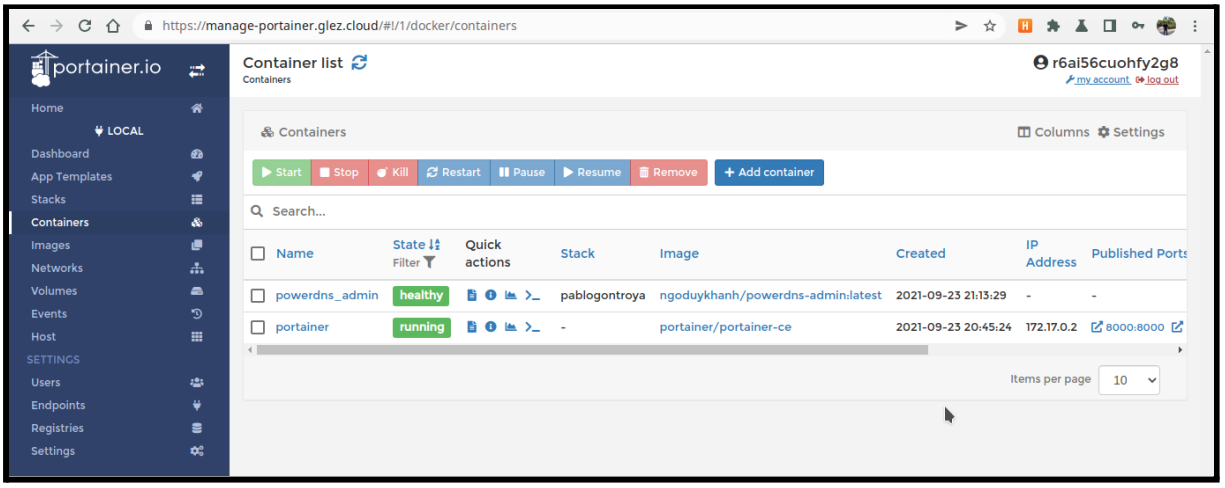
5.1.7.5. Comprobación de funcionamiento y ventajas

La gran ventaja de este servicio es que no es necesario abrir puertos en el firewall, pues es el agente de Cloudflare quien inicia la conexión hacia el centro de datos de Cloudflare (IN > OUT). Es en el centro de datos de Cloudflare donde se termina la conexión del cliente.

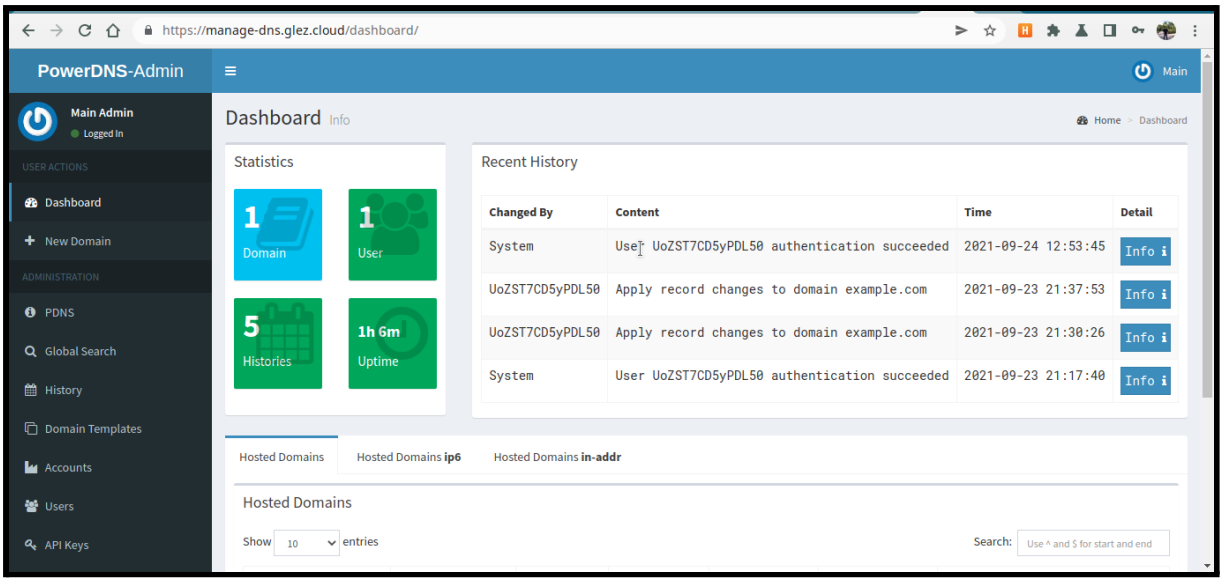
Respecto a la página de estadísticas, podemos ver como es segura y no ha sido necesario indicar el puerto especial. De hecho, ni siquiera está abierto en el Firewall.



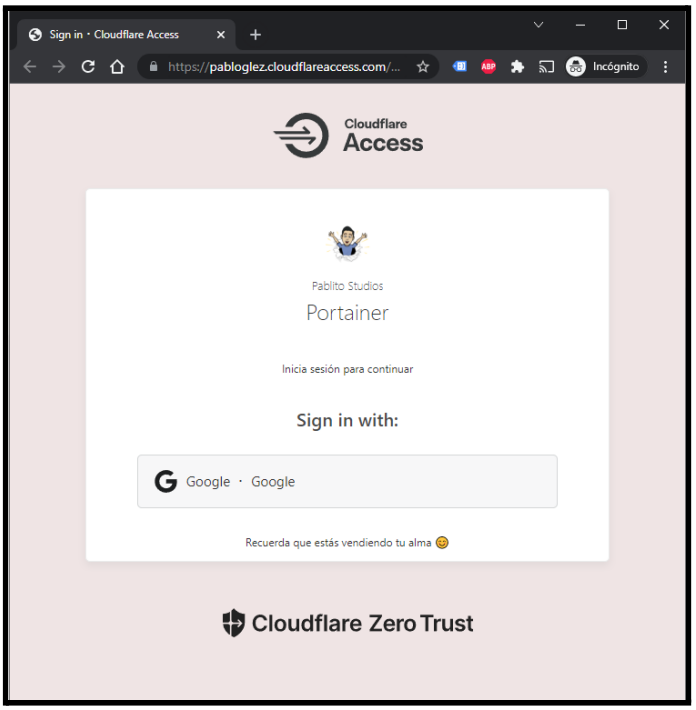
Respecto a la interfaz de Portainer:



Y por último, la interfaz de gestión web de los registros DNS:



También se puede combinar con la necesidad de autenticación antes de acceder al servicio. Por ejemplo, con una cuenta de Google, Microsoft o Facebook. O un código enviado al email. Podemos hacer que solo ciertas cuentas tengan acceso a ciertas páginas que no deben ser públicas (como la parte de estadísticas de OpenDNS), creando para ello una lista de identidades permitidas.



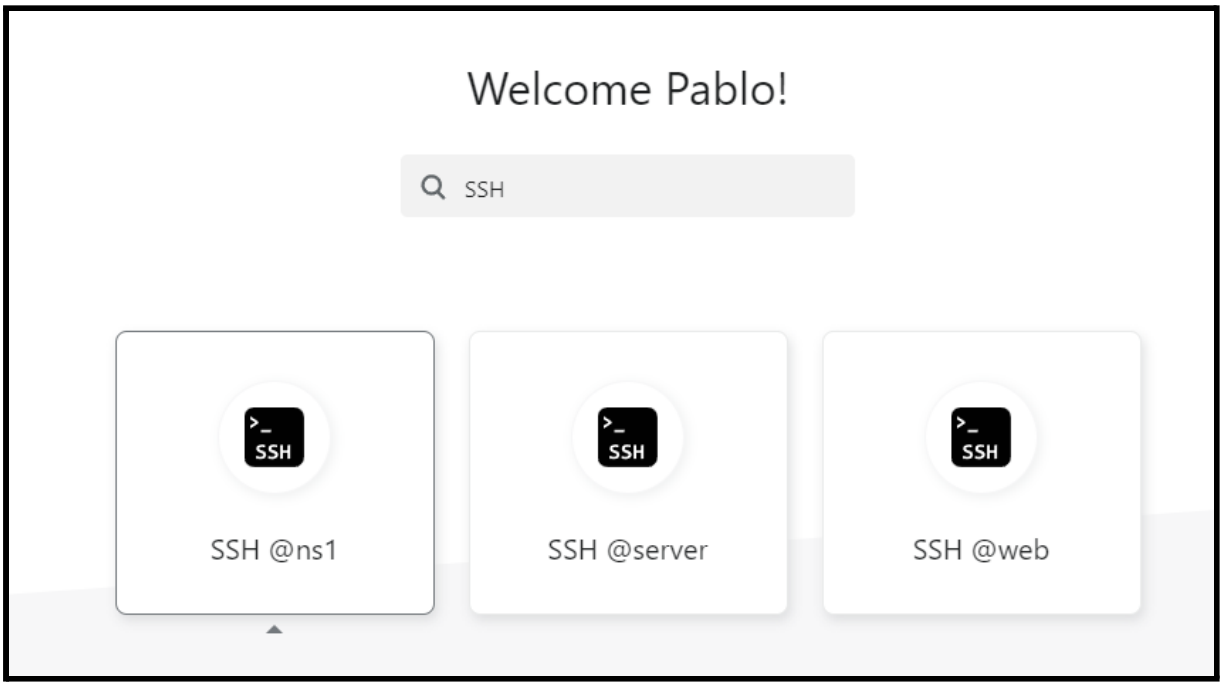
5.1.7.6. Servicio SSH en el navegador

De forma adicional al acceso a servicios HTTP a través del proxy inverso de Cloudflare Zero Trust, esta herramienta es capaz de renderizar una sesión SSH en el navegador.

Esta funcionalidad es realmente útil pues debido a una serie de cuestiones:

- No es necesaria la apertura del puerto 22/TCP, por defecto, o cualquier otro. La conexión, como ya se ha comentado, es en sentido servidor → Cloudflare.
- No es necesaria la gestión de claves SSH por parte de los usuarios, lo que elimina el punto de fallo.
- Permite la gestión centralizada de identidades, que bien podría hacerse mediante LDAP o Google Workspace en este caso. Los permisos son modificados de forma centralizada, por lo que como administradores nos es indiferente cuántos servidores tengamos.

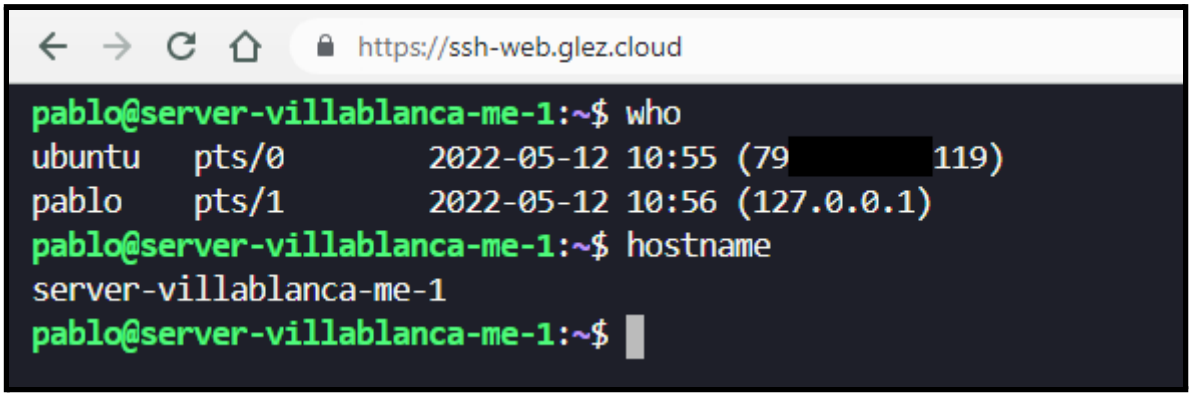
Para acceder basta con acceder a la página de login y buscar SSH. Se mostrarán los resultados en la página:



En este caso, el servidor deseado es *SSH @web*. Simplemente haciendo clic en la tarjeta, o pulsando la tecla Intro, comenzará el inicio de sesión. No nos pedirá ni

usuario ni contraseña. Este dato es, en mi opinión, realmente interesante. Se basa en la emisión de certificados de corta duración, que comentaremos a continuación.

En tanto a la sesión SSH, podemos ver como es una sesión equivalente a la que generaríamos usando Putty o el cliente SSH integrado en Linux:



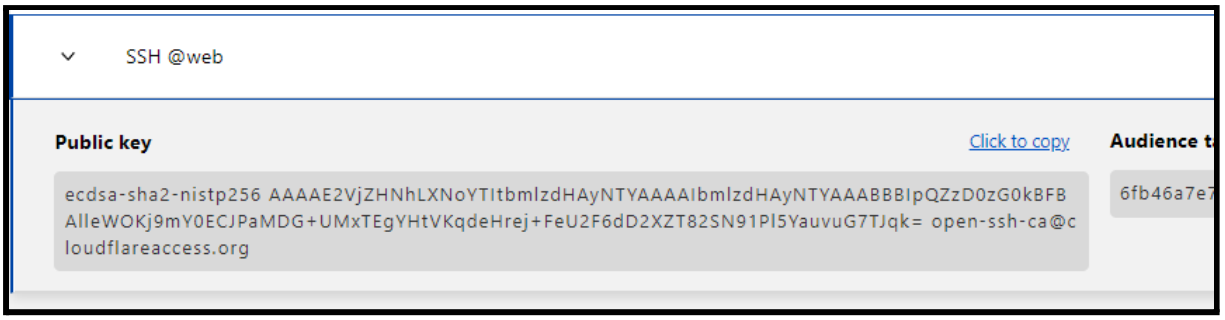
Uso de certificados de corta vida

En el modo tradicional, se generaría un par de claves y el usuario lo mantendría durante años. En este modelo, añadimos la clave pública de una Autoridad de Certificados (CA) virtual administrada por Cloudflare a la lista de confiables. De esta forma, Cloudflare rotará los certificados de forma frecuente y gestionará la autenticación contra el servidor SSH.

Es importante que el usuario en el proveedor de identidades (IdP, *Identity Provider*) se corresponda con el usuario SSH en el servidor. Si mi usuario en el IdP es pepito@ejemplo.es. Mi usuario en el servidor web ha de ser también pepito. De no ser así, no se podrá realizar el login mediante este método. La web nos solicitará indicar usuario y contraseña local en el servidor. Esta forma de autenticación es menos recomendada, pues no es posible aprovechar todas las ventajas ofrecidas por los *short-lived certificates*.

Una vez verificado que los usuarios se corresponden entre el IdP y el equipo local, debemos generar la clave pública del certificado desde la interfaz web de gestión de Cloudflare Zero Trust. Indicaremos la aplicación para la que queremos general la clave pública y en unos pocos segundos se nos mostrará por pantalla la clave. Esta

la debemos copiar en un archivo de nuestro equipo, al que luego referenciaremos desde los ajustes del servidor SSH:



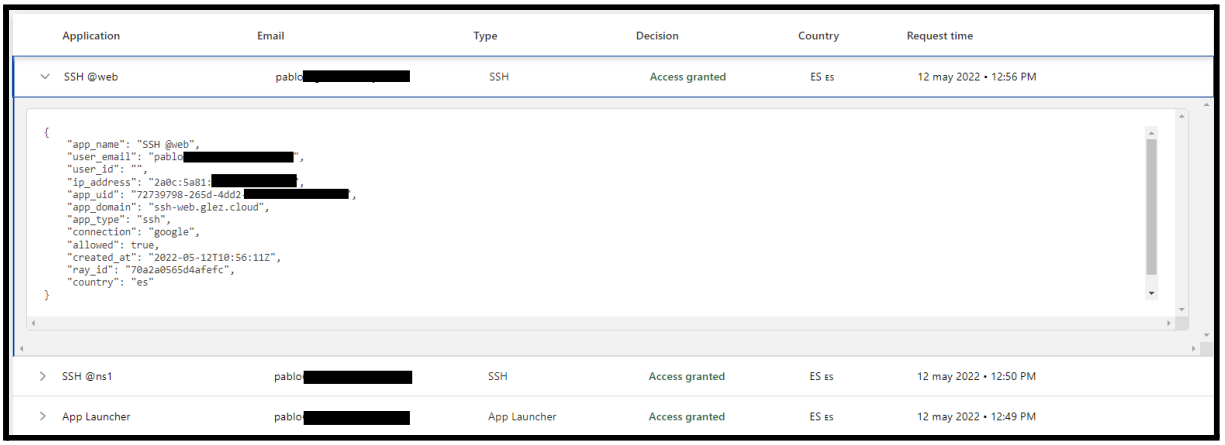
Es recomendable que este archivo no esté en un directorio personal de algún usuario, para evitar problemas con permisos y eliminaciones accidentales.

Una vez tenemos la clave pública en un archivo, procedemos a editar el archivo de configuración del servidor SSH para añadir la referencia al archivo anterior y comprobar como el acceso mediante par de claves está habilitado:

/etc/ssh/sshd_config

```
PubkeyAuthentication yes
TrustedUserCAKeys /etc/ssh/ca.pub
```

De forma previa a la conexión SSH, se ha comprobado como el usuario es quien dice ser (Autenticación) y efectivamente tiene permisos para poder acceder al servicio solicitado (Autorización). Todas las acciones se reflejan en el registro (*Accountability*).



5.2. Servicio e-mail & SMTP 📧

5.2.1. Instalación y problemática 25/TCP en GCP

5.2.1.1. El puerto 25/TCP es bloqueado en GCP: Problema

Debido al riesgo de abuso, las conexiones al puerto TCP de destino 25 siempre están bloqueadas cuando el destino es externo a tu red de VPC.

Esta es la información que proporciona Google respecto al bloqueo aplicado en Cloud Platform[54].

¿Cómo se aplica esta restricción a nuestro particular? No vamos a poder disponer de un sistema de correo electrónico completo. Sí, las interfaces web de gestión están disponibles y aparentemente los mensajes son enviados desde las cuentas hospedadas en el servidor. Pero no salen hacia el exterior de la red en Google Cloud Platform. Como sabemos, el puerto 25 es imprescindible para el envío de correo electrónico. Aunque se *upgradee* la conexión a una cifrada, por defecto el primer HELLO siempre va a ser por el puerto 25.

Comprobando los registros podemos ver errores timed out. Es decir, se agota el tiempo de respuesta del servidor. Esto en el lado del servidor[55]:

Message size	Sender	Recipients	Action
1.4 KiB	info@xn--ahorrams-fza.com	pepe@glez.cloud (connect to mailcow.gcp.glez.cloud[104.199.97.243]:25: Connection timed out)	Show message
962 B	user@glez.cloud	[redacted] (connect to alt2.gmail-smtp-in.l.google.com[2a00:1450:4010:c1c::1b]:25: Network is unreachable)	Show message
1.6 KiB	user@glez.cloud	info@xn--ahorrams-fza.com (connect to mail.xn--ahorrams-fza.com[82.98.134.111]:25: Connection timed out)	Show message
876 B	user@glez.cloud	test@glez.tk (connect to alt1.aspmx.l.google.com[142.251.9.27]:25: Connection timed out) [redacted] connect to alt2.gmail-smtp-in.l.google.com[142.250.150.26]:25: Connection timed out)	Show message

[54] https://cloud.google.com/compute/docs/tutorials/sending-mail
[55] La imagen es en GCP, aunque la instalación recogida en este documento se haya hecho en OVH.

Aunque aquí no lo podemos ver, hay una serie de mensajes que sí han llegado a la bandeja de entrada: los enviados desde info@ahorramás.com hacia admin@glez.cloud. Llegan porque los servidores negocian directamente en puertos diferentes al 25. Son también puertos estándar como el 587 o 465, que sirven para el intercambio de mensajes entre Mail User Agents, MUA, utilizando SSL y TLS.

5.2.1.2. El puerto 25/TCP es bloqueado en GCP: Solución

En el artículo de Google donde se anuncia el bloqueo nos proponen dos soluciones, que sí pueden ser útiles para otros clientes. Citando dicho artículo:

> *SendGrid, Mailgun y Mailjet ofrecen un nivel gratuito para que los clientes de Compute Engine puedan configurar y enviar correos electrónicos a través de sus servidores. Si no tienes una cuenta de Google Workspace, usa estos socios externos para aprovechar funciones como el seguimiento de clics, las estadísticas, las API y otras funciones a fin de satisfacer tus necesidades de correo electrónico.*
>
> *Como alternativa, si estás familiarizado con Google Workspace y ya pagas por una cuenta del producto que admite correo electrónico, puedes configurar un servicio de retransmisión para enviar correos electrónicos a través de Google Workspace. Ten en cuenta que Gmail y Google Workspace aplican límites a la actividad de correo electrónico.*

Estudiando las posibilidades podemos comentar:

- La primera opción no se nos aplica al completo, pues uno de los objetivos del Proyecto Fin de Grado es gestionar de forma autónoma un sistema completo de envío de correo electrónico. Si delegamos parte del servicio a otra empresa, perdemos alcance. También cabe estacar el coste de estos servicios
- En tanto a la segunda opción, sí que cabría la posibilidad de utilizar los servicios de Relay SMTP que Google Workspace ofrece a sus clientes. Pero, al igual que con la propuesta anterior, perderíamos alcance en tanto al objetivo del proyecto de fin de grado.

  También cabe destacar que Google impone ciertos límites en el envío de correo (no pretende que su plataforma se dedique a esta función, sino a la

colaboración). También habría que añadir cada dominio desde la Consola de Administración de Google.

Ninguna de las opciones a las que Google nos dirige son completamente aplicables a nuestro caso. Por tanto, se decide probar en otras nubes públicas. No estamos buscando soluciones con costes fijos mensuales ni compromisos de permanencia. Esto deja fuera muchas opciones, incluidas la mayoría de las empresas locales que no terminan de adoptar el modelo de nube pública. Sin embargo, sí hay una empresa local que destaca en este sentido y es clouding.io.

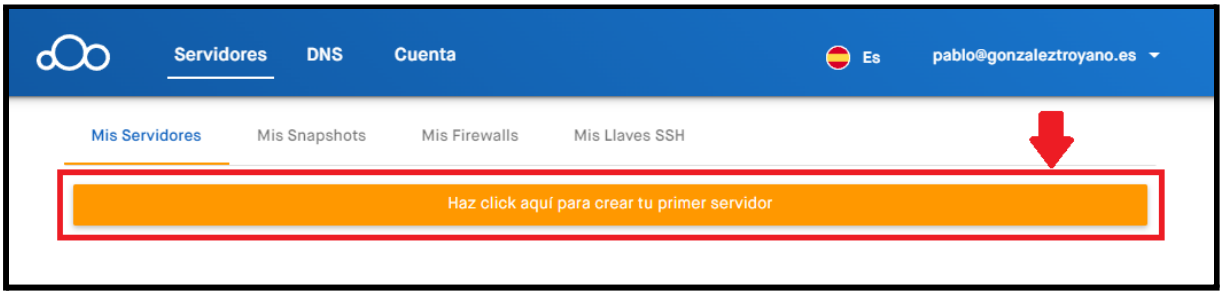
Para esta sección del proyecto se ha alojado el servidor en clouding.io, que ha colaborado con el proyecto patrocinando los servidores utilizados.

Por tanto, el servidor virtual de correo electrónico se alojará en este proveedor.

5.2.1.3. Definición del servidor virtual

La interfaz de gestión de Clouding es muy sencilla. Vamos a ver el procedimiento de creación paso a paso.

Una vez iniciada la sesión en nuestra cuenta, debemos hacer clic en este botón:



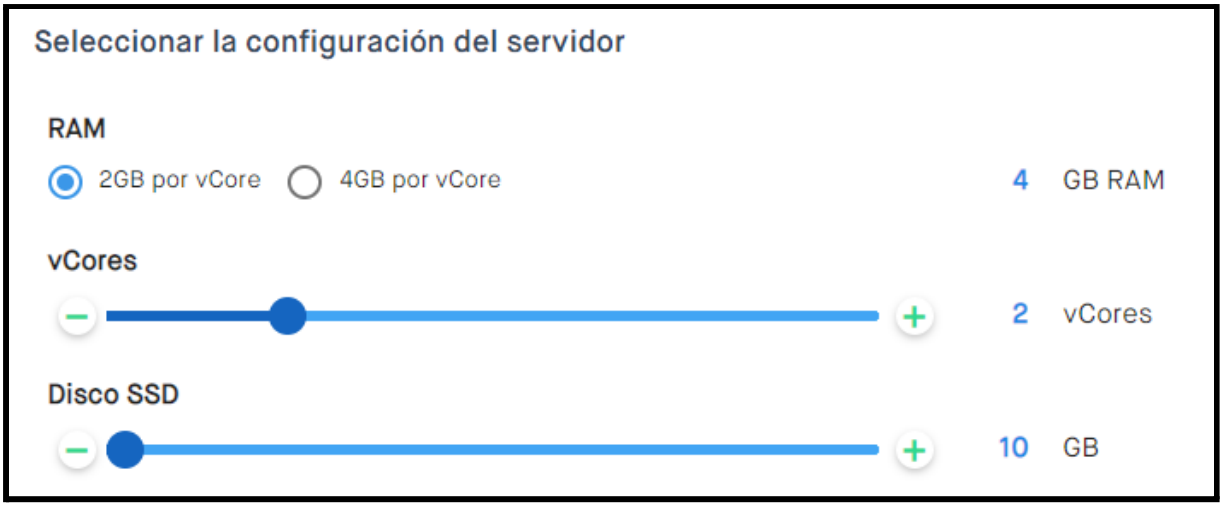
Indicamos un nombre del servidor, en nuestro caso hemos definido *mail.glez.cloud*. También debemos seleccionar el origen del disco, en este caso hemos seleccionado una imagen Ubuntu 18.04:

También debemos seleccionar la cantidad de memoria RAM y vCores que asignaremos a nuestra máquina.

En nuestro caso, seleccionamos 4 GB de RAM, 2 vCores y 10 GB de disco SSD. Realmente puede ser incluso demasiado para la implementación, pero así nos aseguramos de que “no nos quedamos cortos”, al menos durante la implantación.

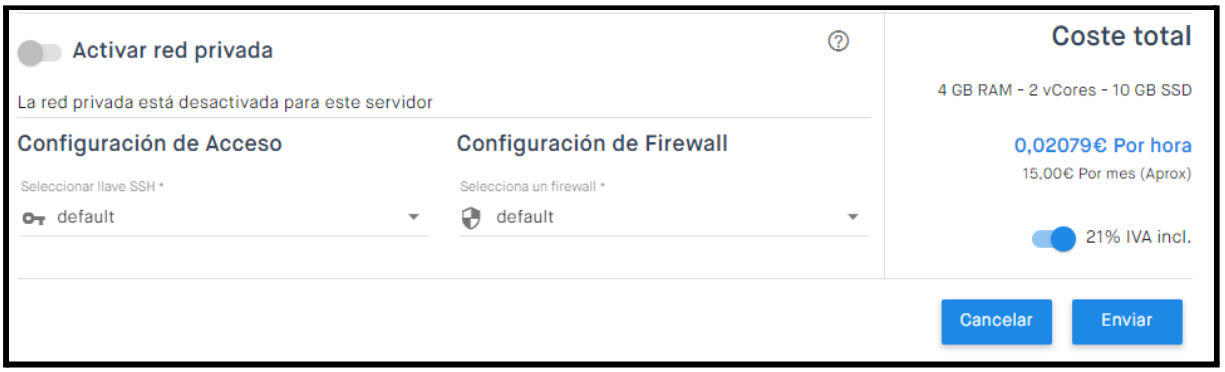
Gracias a las ventajas que nos aporta la nube pública, podemos ampliar y reducir la capacidad de computación según sea necesario por las aplicaciones y usuarios.

En la siguiente imagen podemos observar la configuración comentada anteriormente en la interfaz de gestión web de Clouding:

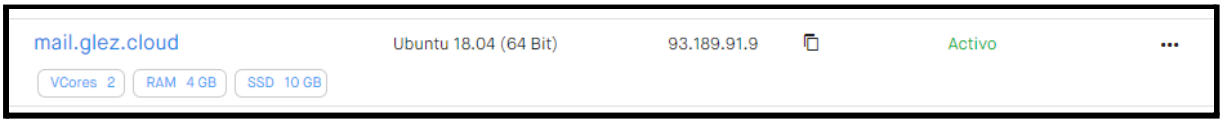


También, por seguridad, se van a activar los backups del disco de la MV. Se realizará un *snapshot* cada dos días y se guardarán los 7 últimos, por tanto podremos recuperar el estado anterior durante 14 días:

Seleccionamos la llave SSH, el ajuste de Firewall y revisamos el precio:



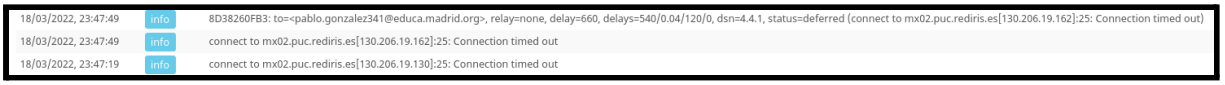
Hacemos clic en *Enviar* para proceder a la creación de la instancia. Ya la vemos disponible en nuestro dashboard:



Por último, debemos tener en cuenta que por seguridad el tráfico SMTP está bloqueado. Digamos que debemos tener un doble interruptor. Por un lado, el firewall en sí, y luego el siguiente interruptor accesible desde nuestro panel de control, pinchando en el servidor deseado, y luego sobre la pestaña *Red*.

Debemos asegurarnos de que ese interruptor está activado. Si no lo estuviera, basta con hacer clic sobre él y luego confirmar la operación en el cuadro de diálogo que nos aparecerá.

De no hacerlo, corremos el riesgo de recibir errores como los siguientes en el registro. Nuestros correos no serán entregados, ni recibiremos los correos que se nos envíen:



5.2.1.4. Creación de registros DNS de infra

Se crean los siguientes registros DNS para el dominio glez.cloud:

```
mailcow.cio.glez.cloud     3600 A        93.189.91.9
mail                       300 CNAME     mailcow.cio.glez.cloud
autodiscover               300 CNAME     mailcow.cio.glez.cloud
autoconfig                 300 CNAME     mailcow.cio.glez.cloud
```

5.2.1.5. Docker: contenedores incluidos e instalación

Todo el despliegue se realiza utilizando contenedores, con un *compose*. Este docker-compose tiene los siguientes contenedores:

- ACME - https://letsencrypt.org/
- ClamAV - https://www.clamav.net/
- Dovecot - https://www.dovecot.org/
- MariaDB - https://mariadb.org/
- Memcached - https://www.memcached.org/
- Netfilter - https://www.netfilter.org/
- Nginx - https://nginx.org/
- Oletools - https://github.com/decalage2/oletools
- PHP - https://php.net/
- Postfix - http://www.postfix.org/
- Redis - https://redis.io/
- Rspamd - https://www.rspamd.com/
- SOGo - https://sogo.nu/
- Solr - https://solr.apache.org/
- Unbound - https://unbound.net/
- Watchdog

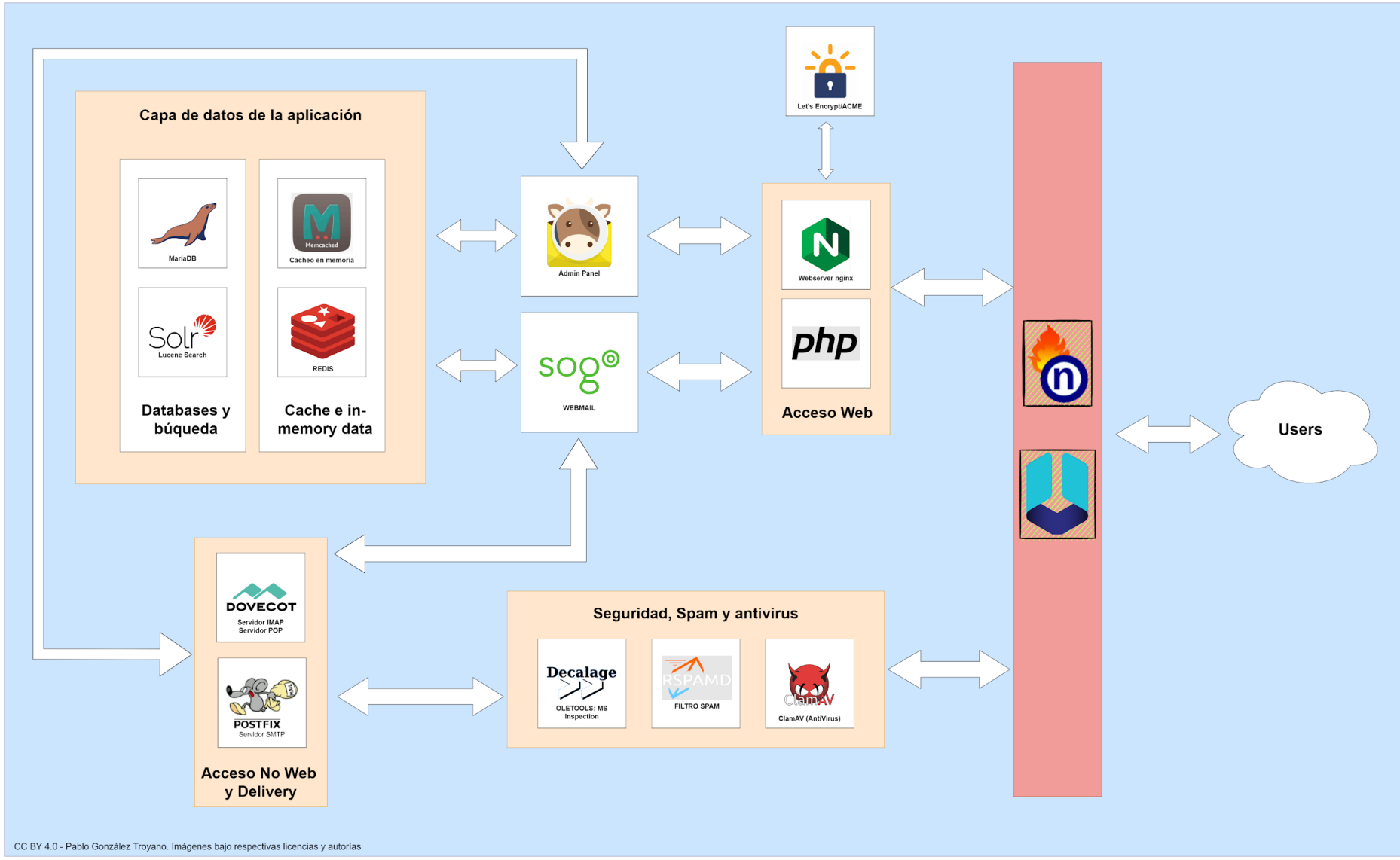
Instalamos docker:

```
curl -sSL https://get.docker.com/ | CHANNEL=stable sh
systemctl enable --now docker
```

Instalamos también docker-compose:

```
sudo su

curl -L https://github.com/docker/compose/releases/download/$(curl -Ls
https://www.servercow.de/docker-compose/latest.php)/docker-compose
-$(uname -s)-$(uname -m) > /usr/local/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

exit
```

Para comprender mejor la relación entre los distintos componentes de mailcow dockerizado, lo mejor es revisar el siguiente esquema de arquitectura. No es completamente exhaustivo ni técnico, pues su objetivo es comprender a grandes rasgos los componentes.


Capa de datos de la aplicación
MariaDB
Memcached
Cacheo en memoria
Solr
Lucene Search
REDIS
Databases y búqueda
Cache e in-memory data
Let's Encrypt/ACME
Admin Panel
sogo
WEBMAIL
Webserver nginx
php
Acceso Web
Users
DOVECOT
Servidor IMAP
Servidor POP
POSTFIX
Servidor SMTP
Acceso No Web y Delivery
Seguridad, Spam y antivirus
Decalage
OLETOOLS: MS Inspection
RSPAMD
FILTRO SPAM
ClamAV
ClamAV (AntiVirus)
CC BY 4.0 - Pablo González Troyano. Imágenes bajo respectivas licencias y autorías

5.2.1.6. Docker: clone del repositorio y script de configuración

Verificamos el umask de nuestro repositorio, que tiene que ser 0022. Para hacerlo, ejecutamos:

```
umask
```

A continuación, nos situamos en la ruta de destino correcta y clonamos el repositorio, para situarnos sobre el directorio de destino:

```
cd /opt
git clone https://github.com/mailcow/mailcow-dockerized
cd mailcow-dockerized
```

Ejecutamos el siguiente script de autoconfiguración. Debemos tener en cuenta que la parte más importante es la del FQDN de la máquina (por el certificado):

```
./generate_config.sh
```

Antes de introducirlo en el propio script es altamente recomendable realizar una consulta dig para comprobar que la IP devuelta es la deseada:

```
dig mail.glez.cloud +short @1.1.1.1
```

```
pablo@WIN-PABLO:~$ dig mail.glez.cloud +short @1.1.1.1
mailcow.cio.glez.cloud.
93.189.91.9
```

Se habrá generado un archivo de configuración, `mailcow.conf`, que podemos consultar de forma sencilla con el siguiente comando:

```
grep -v -e "#" mailcow.conf | grep -v -e "^$"
```

El archivo estará disponible en el repositorio de GitHub[56] y en el anexo VI de este documento (*Anexo VI: Códigos relativos al servicio de correo*).

[56] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/2-mail/mailcow.conf.txt

Ahora que ya tenemos todo configurado, basta con conseguir las imágenes de los contenedores y lanzarlos desde el archivo docker-compose.yml (disponible en el repositorio de GitHub[57]):

```
docker-compose pull
docker-compose up -d
```

```
root@mail:/opt/mailcow-dockerized# docker-compose pull
Pulling unbound-mailcow   ...
Pulling mysql-mailcow     ... done
Pulling redis-mailcow     ... done
Pulling clamd-mailcow     ...
Pulling php-fpm-mailcow   ...
Pulling sogo-mailcow      ... extracting (99.5%)
Pulling dovecot-mailcow   ... done
Pulling rspamd-mailcow    ... done
Pulling postfix-mailcow   ...
Pulling memcached-mailcow ... extracting (100.0%)
Pulling nginx-mailcow     ...
Pulling acme-mailcow      ...
Pulling netfilter-mailcow ...
Pulling watchdog-mailcow  ... extracting (71.1%)
Pulling dockerapi-mailcow ... done
Pulling solr-mailcow      ... downloading (84.1%)
Pulling olefy-mailcow     ...
Pulling ofelia-mailcow    ... done
Pulling ipv6nat-mailcow   ...
```

```
root@mail:/opt/mailcow-dockerized# docker-compose up -d
Creating network "mailcowdockerized_mailcow-network" with driver "bridge"
Creating volume "mailcowdockerized_vmail-vol-1" with default driver
Creating volume "mailcowdockerized_vmail-index-vol-1" with default driver
Creating volume "mailcowdockerized_mysql-vol-1" with default driver
Creating volume "mailcowdockerized_mysql-socket-vol-1" with default driver
Creating volume "mailcowdockerized_redis-vol-1" with default driver
Creating volume "mailcowdockerized_rspamd-vol-1" with default driver
Creating volume "mailcowdockerized_solr-vol-1" with default driver
Creating volume "mailcowdockerized_postfix-vol-1" with default driver
Creating volume "mailcowdockerized_crypt-vol-1" with default driver
Creating volume "mailcowdockerized_sogo-web-vol-1" with default driver
Creating volume "mailcowdockerized_sogo-userdata-backup-vol-1" with default driver
Creating mailcowdockerized_memcached-mailcow_1 ...
Creating mailcowdockerized_clamd-mailcow_1     ...
Creating mailcowdockerized_watchdog-mailcow_1  ...
Creating mailcowdockerized_sogo-mailcow_1      ...
Creating mailcowdockerized_solr-mailcow_1      ...
```

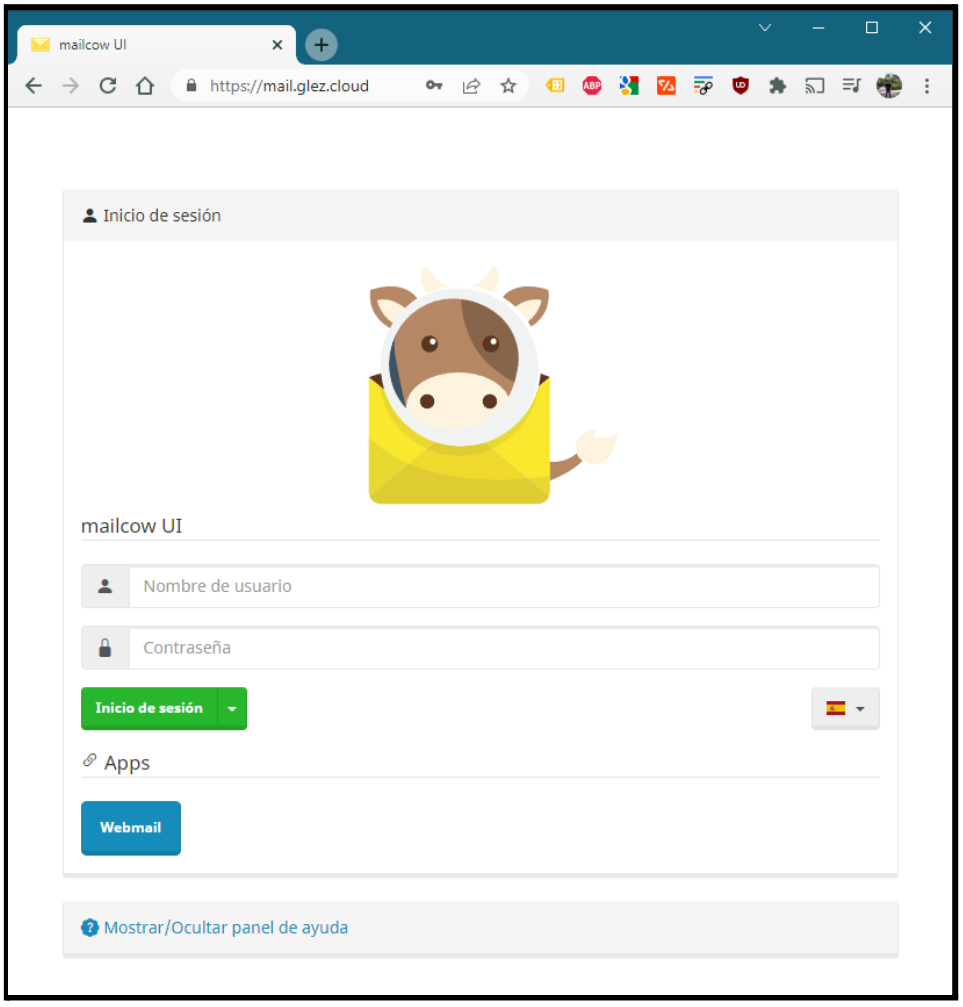
[57] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/2-mail/docker-compose.yml

5.2.1.7. Primer acceso a la interfaz web y redirección HTTPS

Pasados unos segundos, la interfaz web estará disponible en el FQDN indicado, de ver algún error podemos ejecutar:

```
docker-compose logs --tail=200 php-fpm-mailcow nginx-mailcow
```

Si ahora accedemos a https://mail.glez.cloud veremos la pantalla de inicio de Mailcow:



Las credenciales por defecto son `admin`:`moohoo`.

Como nos habremos dado cuenta, las peticiones HTTP no son redireccionadas a HTTPS, algo que supone un gran fallo de seguridad[58]. Además, las redirecciones 301 hacía HTTPS son beneficiosas desde el punto de vista SEO, puesto que los

[58] https://mailcow.github.io/mailcow-dockerized-docs/manual-guides/u_e-80_to_443/

buscadores hoy en día premian esta seguridad extra ofrecida por los certificados SSL/TLS.

Para corregirlo vamos a crear un nuevo archivo en la ruta `data/conf/nginx/redirect.conf`, en el que incluiremos el siguiente contenido:

```
server {
  root /web;
  listen 80 default_server;
  listen [::]:80 default_server;
  include /etc/nginx/conf.d/server_name.active;
  if ( $request_uri ~* "%0A|%0D" ) { return 403; }
  location ^~ /.well-known/acme-challenge/ {
    allow all;
    default_type "text/plain";
  }
  location / {
    return 301 https://$host$uri$is_args$args;
  }
}
```

Una vez editado y guardado el archivo anterior, basta con reiniciar el contenedor de nginx:

```
docker-compose restart nginx-mailcow
```

5.2.2. Configuración inicial de Mailcow

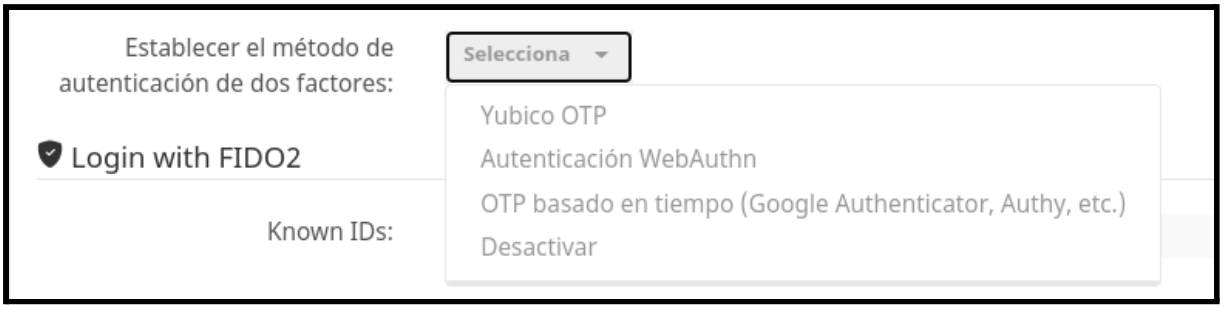
5.2.2.1. Desactivación usuario *admin* y MFA

Lo primero que debemos hacer según accedemos a la instancia es deshabilitar el usuario administrador y crear uno con distinto nombre de usuario y una contraseña compleja. También es altamente recomendable implementar la verificación en dos pasos con una llave de seguridad.

Como siempre, la contraseña quedará reflejada en Anexo III: Contraseñas de los servicios.

Una vez cambiada la contraseña, volvemos a iniciar la sesión. También activaremos la verificación en dos pasos para añadir una capa de seguridad adicional a nuestra cuenta.

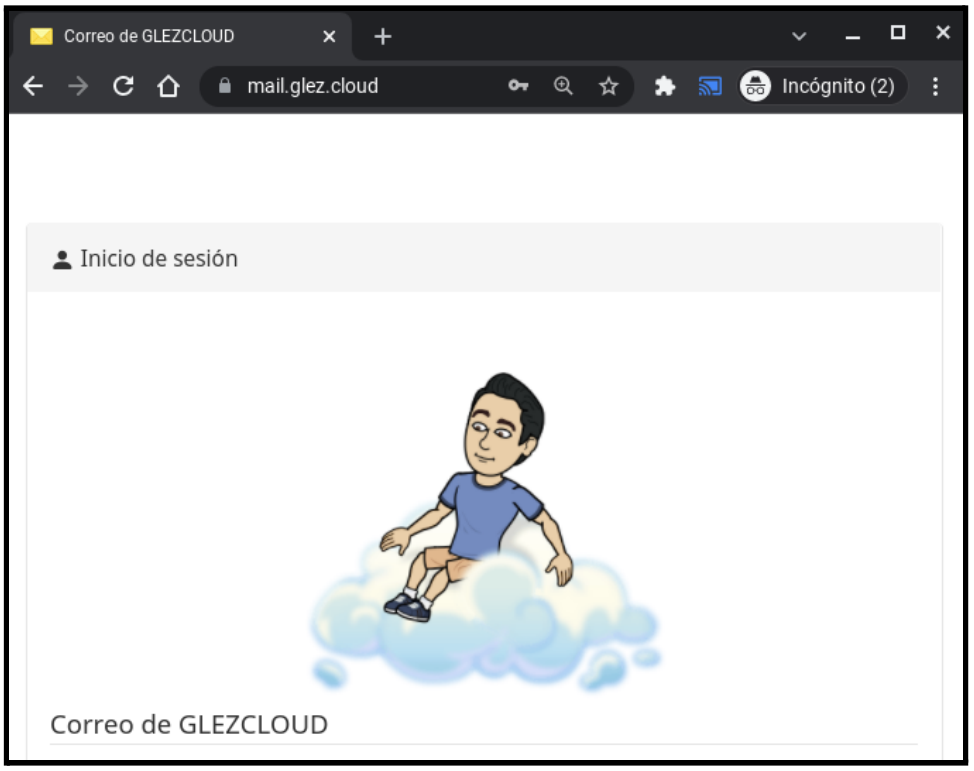
Seleccionamos, en la pantalla principal, la opción que decidamos:



En nuestro caso, se ha elegido *OTP basado en tiempo*, utilizando la aplicación de Authy.

5.2.2.2. Personalización de la interfaz

Se eliminarán las alusiones a *Mailcow* para sustituirlas por *GLEZCLOUD*. Esta operación se realiza desde el menú de operaciones > Configuración > Personalización. Así es como ha quedado:



5.2.2.3. Adición de un dominio a la interfaz. Añadir DKIM.

Vamos a proceder a añadir el dominio carpet4you.site al sistema de correo. Para acceder a la interfaz, hacemos clic en *Configuración*, después en *Buzones*. En esta pantalla, hacemos clic en *+ Agregar dominio*.

Insertamos *carpet4you.site* en el campo del dominio. Dejamos los valores por defecto, asegurando de que hemos seleccionado 1024 bits de longitud de clave DKIM (se selecciona 1024 en vez de 2048 puesto que con algunos clientes DNS y de correo puede dar problemas la longitud de 2048). Como selector DKIM indicamos *glezcloud*. Para terminar y guardar los cambios hacemos clic en *Agregar dominio y reiniciar SOGo.*

Para ver ahora los registros relativos al DKIM, nos dirigimos a la sección Administración > Configuración > Registros ARC/DKIM debemos localizar el registro DKIM creado para nuestro nuevo dominio. Aquí veremos el registro TXT que debemos de añadir al DNS.

En relación a DKIM, SPF y DMARC, se recomienda la consulta del Anexo IX.

Para comprobar si se ha propagado correctamente, podemos utilizar el siguiente comando:

```
dig TXT glezcloud._domainkey.carpet4you.site
```

El resultado será similar al siguiente:

```
; <<>> DiG 9.11.5-P4-5.1+deb10u7-Debian <<>> TXT glezcloud._domainkey.carpet4you.site
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4012
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;glezcloud.carpet4you.site.     IN      TXT

;; ANSWER SECTION:
glezcloud._domainkey.carpet4you.site. 3600 IN      TXT
"v=DKIM1;k=rsa;t=s;s=email;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDiyYwTn2MzbP7nw1dNFBl
D9draWae/SZ2FNyQjynlqdjnluFFmfLuSFaFDG9Rs67et8tYqz2YDGNWB89U+rabGM+wO+tJ+hd5799junhf0IHf
z4RfvYunLHMTebyg83MLvlFh3jQwHsLBR5voB40sAiQPw8sLTNOhId4FbRmYJGwIDAQAB"

;; Query time: 38 msec
;; SERVER: 100.115.92.193#53(100.115.92.193)
;; WHEN: Fri Mar 18 22:36:26 CET 2022
;; MSG SIZE  rcvd: 311
```

También añadiremos el siguiente registro MX:

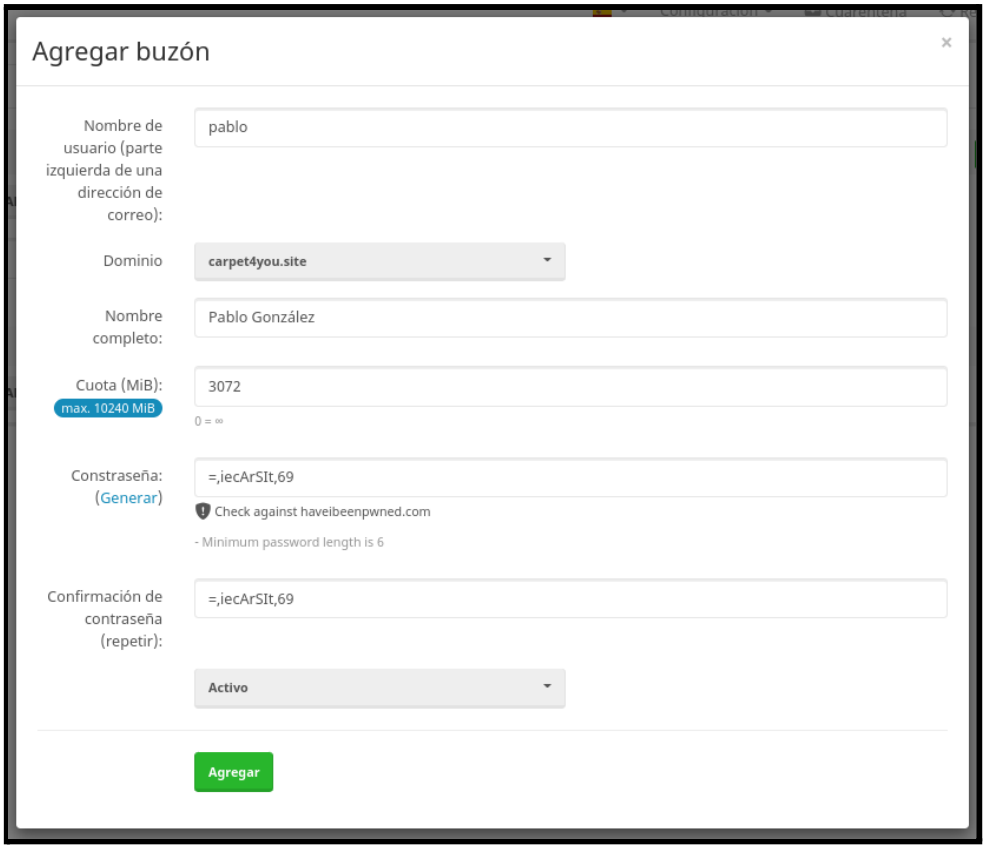
En tanto al SPF:

```
60 IN TXT "v=spf1 mx -all"
```

El proceso de creación de un usuario nuevo es altamente sencillo. Basta con indicar el nombre y el dominio deseado.

Nos generará una contraseña de forma automática, si bien podemos indicar nosotros la deseada. La generada se ha incluido en el anexo III de este documento como referencia. También nos ofrecerá la posibilidad de limitar la capacidad máxima del buzón. Introduciremos un nombre visible para el nuevo usuario creado.

En la siguiente captura de pantalla podemos ver todos los campos rellenos:

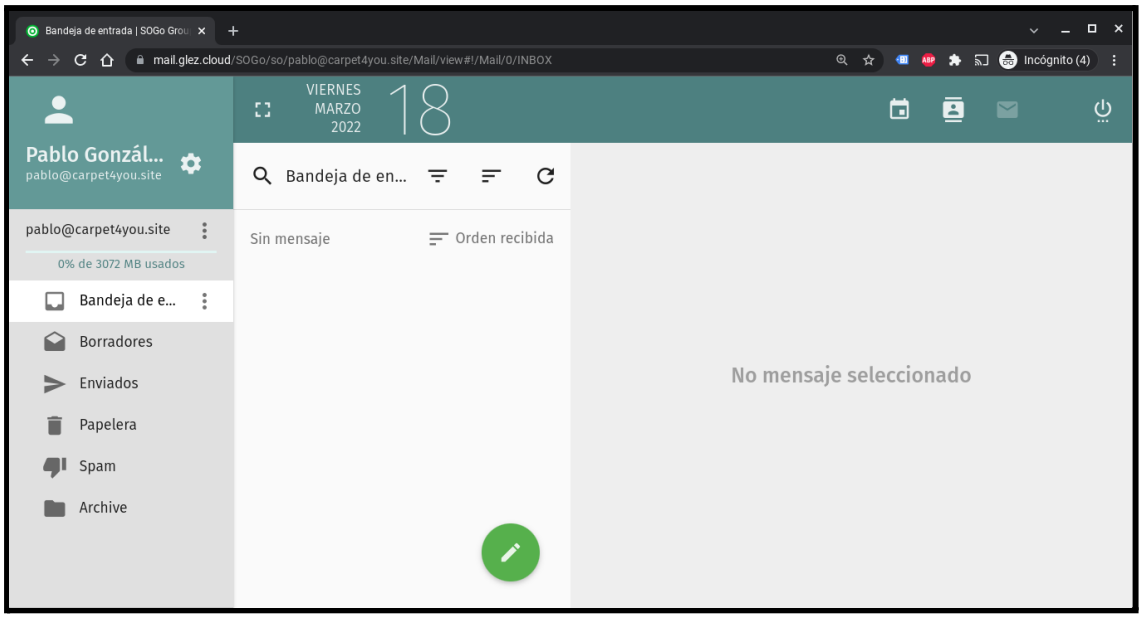


Para iniciar la sesión en el webmail, el usuario debe navegar hasta https://mail.glez.cloud/SOGo/. Esto es lo que verá:

Basta con introducir correo electrónico y la contraseña generada en el paso anterior y hacer clic en la flecha para entrar.

Si la contraseña introducida es correcta, nos saludará con un “Bienvenido Pablo González”.

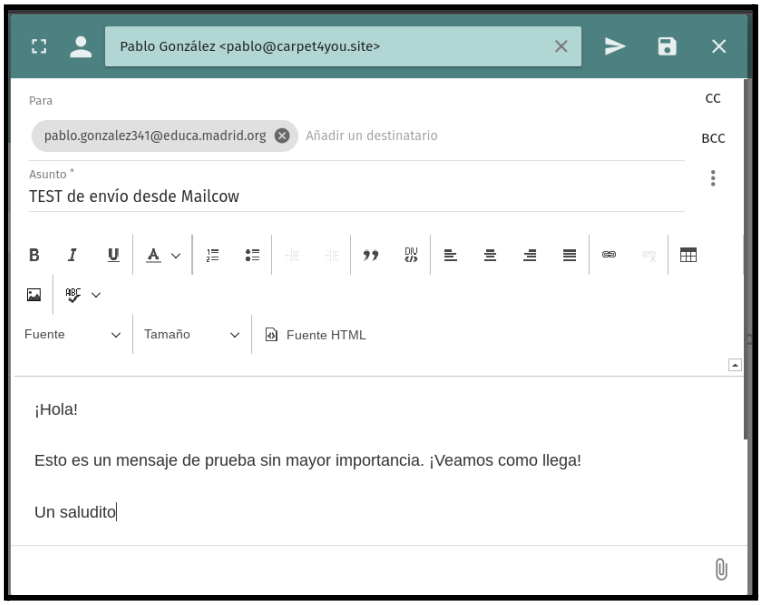
Esta es la pantalla principal de la interfaz web:



5.2.3. Pruebas de envío de correo electrónico

5.2.3.1. Con origen el servidor de correo

Redactemos un nuevo mensaje, como haríamos con cualquier otro cliente de correo electrónico:



En este caso, el destinatario del correo electrónico es mi usuario en EducaMadrid. Estos son los registros desde el servidor de correo:

21/03/2022, 18:56:53	info	DC07C6E8E8: removed
21/03/2022, 18:56:53	info	DC07C6E8E8: to=<pablo.gonzalez341@educa.madrid.org>, relay=mx01.puc.rediris.es[130.206.19.162]:25, delay=2.3, delays=1/0.05/0.49/0.69, dsn=2.0.0, status=sent (250 2.0.0 22LHuq3P022464-22LHuq3R022464 Message accepted for delivery)
21/03/2022, 18:56:52	info	Trusted TLS connection established to mx01.puc.rediris.es[130.206.19.162]:25: TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256 bits) key-exchange ECDHE (P-256) server-signature RSA-PSS (2048 bits) server-digest SHA256 client-signature RSA-PSS (4096 bits) client-digest SHA256
21/03/2022, 18:56:52	info	disconnect from mailcowdockerized_sogo-mailcow_1.mailcowdockerized_mailcow-network[172.22.1.248] ehlo=1 auth=1 mail=1 rcpt=1 data=1 quit=1 commands=6
21/03/2022, 18:56:52	info	DC07C6E8E8: from=<pablo@carpet4you.site>, size=1303, nrcpt=1 (queue active)
21/03/2022, 18:56:51	info	DC07C6E8E8: message-id=<3d-6238bc80-3-55672080@75601252>
21/03/2022, 18:56:51	info	DC07C6E8E8: replace: header Received: from 33578185f0ea (mailcowdockerized_sogo-mailcow_1.mailcowdockerized_mailcow-network [172.22.1.248])??(Authenticated sender: pablo@carpet4you.site)??by mail.glez.cloud (Postcow) with ESMTPA from mailcowdockerized_sogo-mailcow_1.mailcowdockerized_mailcow-network[172.22.1.248]; from=<pablo@carpet4you.site> to=<pablo.gonzalez341@educa.madrid.org> proto=ESMTP helo=<33578185f0ea>: Received: from [127.0.0.1] (localhost [127.0.0.1]) by localhost (Mailerdaemon) with ESMTPA id DC07C6E8E8??for <pablo.gonzalez341@educa.madrid.org>; Mon, 21 Mar 2022 18:56:51 +0100 (CET)

Como se puede observar en ellos, el servidor de correo de EducaMadrid (`mx01.puc.rediris.es[130.206.19.162]:25`) ha aceptado el mensaje.

Después de 40 minutos el mensaje no ha llegado, por lo que se da por fallida esta prueba. Sin duda, el problema parece estar del lado de RedIRIS/EducaMadrid y/o de su filtro de Spam. Para comprobarlo, se harán más pruebas.

También usaremos la herramienta online mail-tester.com para comprobar los aspectos de seguridad y filtrado Spam que pudieran estar afectando al envío de correo electrónico. Esta herramienta “leerá” todas las cabeceras del mensaje para darnos una puntuación. Es muy similar a lo que hacen los servidores de correo electrónico a nivel global, pero siendo mucho más transparente. Aquí sí podemos ver motivos de puntuación y recomendaciones para mejorarlas.

Este servicio, mail-tester.com asigna una dirección exclusiva para cada prueba. En nuestro caso es la siguiente:

A la hora de enviar el correo electrónico, es importante que tenga algo de contenido, sino nuestra puntuación se reducirá. En cualquier caso, alojar un servidor de correo hoy en día es complicado. Más aún si va a ser cuestión de unos días o meses, como es nuestro caso. Esto es debido a que las IPs de los servidores (los pequeños y/o nuevos sobre todo) necesitan generarse una buena reputación para que los grandes servicios (Office,Yahoo, Google, 1&1, etc) "confíen" en ellos y no los marquen directamente como correo no deseado.

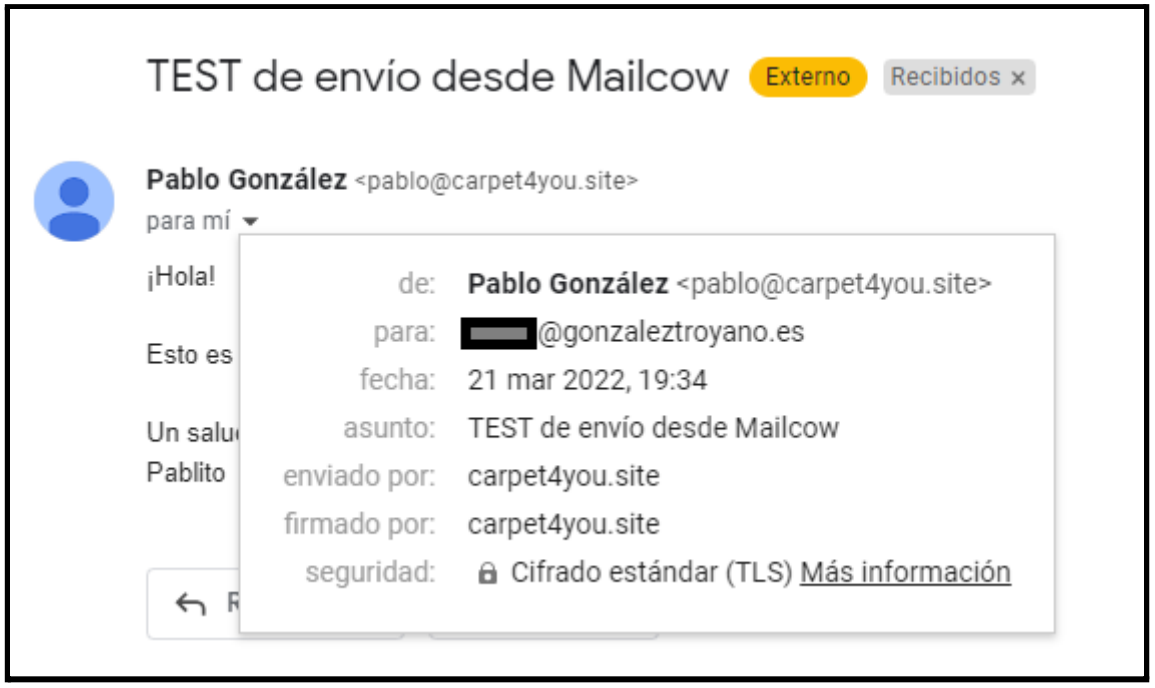
Una vez enviado el correo electrónico a esta herramienta vemos el *report*.También es accesible desde este enlace[59].

Respecto a la puntuación faltante hasta la 10 (perfecta), vemos que es por el TLD (.site). El filtro de spam, SpamAssassin (uno de los más utilizados a nivel mundial), interpreta que este Top Level Domain es utilizando para enviar correos electrónicos fraudulentos:

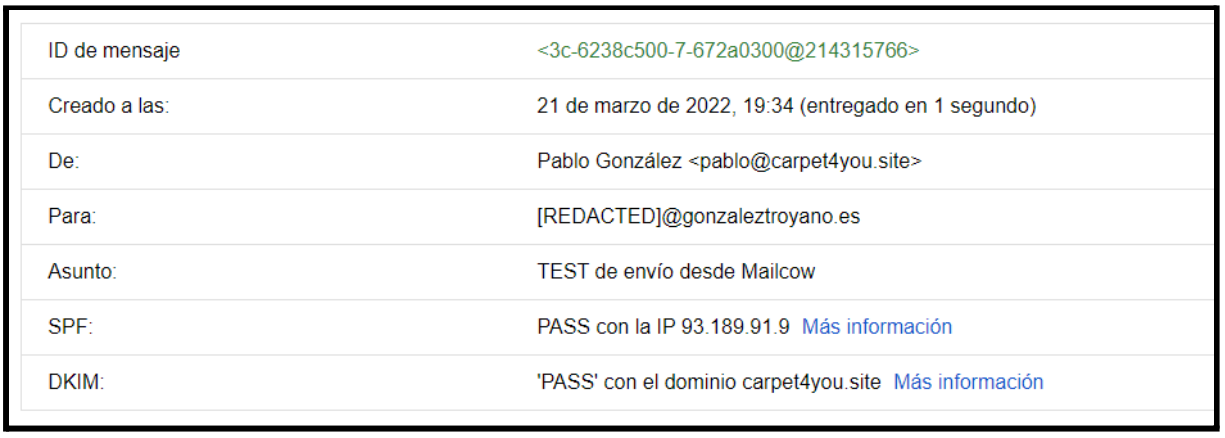
-0.499	FROM_SUSPICIOUS_NTLD	From abused NTLD
-0.001	FROM_SUSPICIOUS_NTLD_FP	From abused NTLD
-0.001	HTML_MESSAGE	HTML included in message **No te preocupes, es normal si envías correos HTML**
-1.725	PDS_OTHER_BAD_TLD	Untrustworthy TLDs URI: carpet4you.site (site)
-0.363	RDNS_DYNAMIC	Delivered to internal network by host with dynamic-looking rDNS
-0.001	SPF_HELO_NONE	SPF: HELO does not publish an SPF Record

[59] https://www.mail-tester.com/test-uq88yxmus

A su vez, se ha probado a enviar un mensaje de correo electrónico a una dirección dentro del dominio gonzaleztroyano.es, cuyo servidor MX es el de Google Workspace. El correo electrónico ha llegado correctamente, si bien ha sido marcado en primera instancia como Spam, lo más probable es que sea debido al TLD.



Como vemos en la imagen anterior, el mensaje de correo electrónico ha sido firmado y ha pasado todos los controles. Antes de investigar las cabeceras del mensaje, que haremos a continuación, vemos como el resumen de seguridad de Google es positivo:



ID de mensaje	<3c-6238c500-7-672a0300@214315766>
Creado a las:	21 de marzo de 2022, 19:34 (entregado en 1 segundo)
De:	Pablo González <pablo@carpet4you.site>
Para:	[REDACTED]@gonzaleztroyano.es
Asunto:	TEST de envío desde Mailcow
SPF:	PASS con la IP 93.189.91.9 Más información
DKIM:	'PASS' con el dominio carpet4you.site Más información

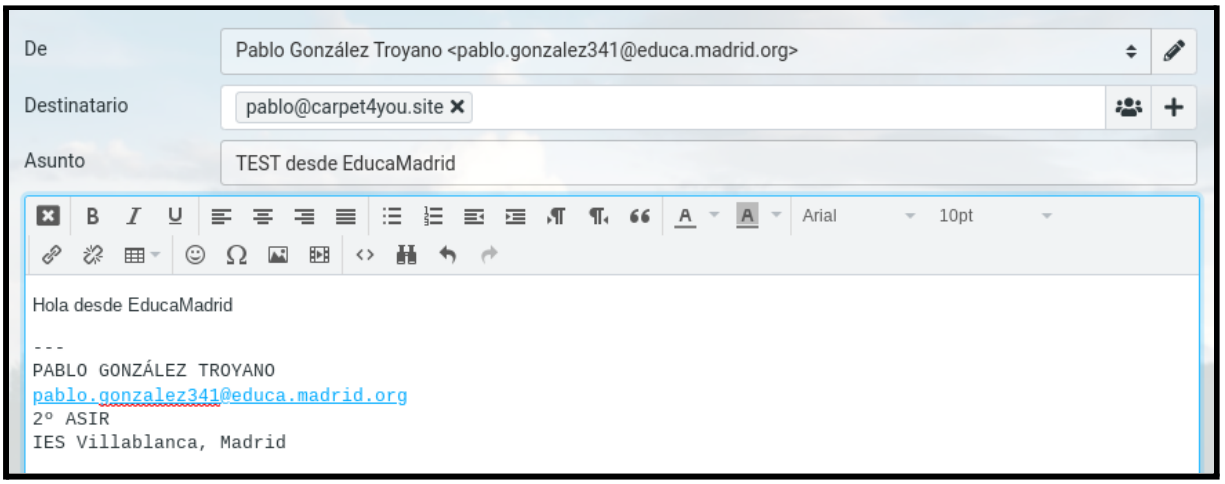
En tanto a las cabeceras del mensaje de correo electrónico, se encuentran disponibles de forma completa en el repositorio de GitHub[60], con la salvedad de que las direcciones de correo electrónico han sido modificadas para prevenir el Spam contra estas.

Veamos algunas partes interesantes de la cabecera, en lo que respecta a SPF y DKIM (puede ser de utilidad la lectura del Anexo IX de este documento):

```
ARC-Authentication-Results: i=1; mx.google.com;
       dkim=pass header.i=@carpet4you.site header.s=glezcloud header.b=DemdnIGl;
       spf=pass (google.com: domain of pablo^^@^^carpet4you.site designates 93.189.91.9ç
       as permitted sender) smtp.mailfrom=pablo^^@^^carpet4you.site

Received: from mail.glez.cloud (de2afb89-06d4-4df5-a344-0d24c913351e.clouding.host. [93.189.91.9])
        by mx.google.com with ESMTPS id
        w9-20020a5d6089000000b00203e9019308si7283664wrt.140.2022.03.21.11.34.29
        for <[REDACTED]^^@^^gonzaleztroyano.es>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Mon, 21 Mar 2022 11:34:29 -0700 (PDT)

Received-SPF: pass (google.com: domain of pablo^^@^^carpet4you.site designates
        93.189.91.9 as permitted sender) client-ip=93.189.91.9;

Authentication-Results: mx.google.com;
       dkim=pass header.i=@carpet4you.site header.s=glezcloud header.b=DemdnIGl;
       spf=pass (google.com: domain of pablo^^@^^carpet4you.site designates 93.189.91.9
       as permitted sender) smtp.mailfrom=pablo^^@^^carpet4you.site
```

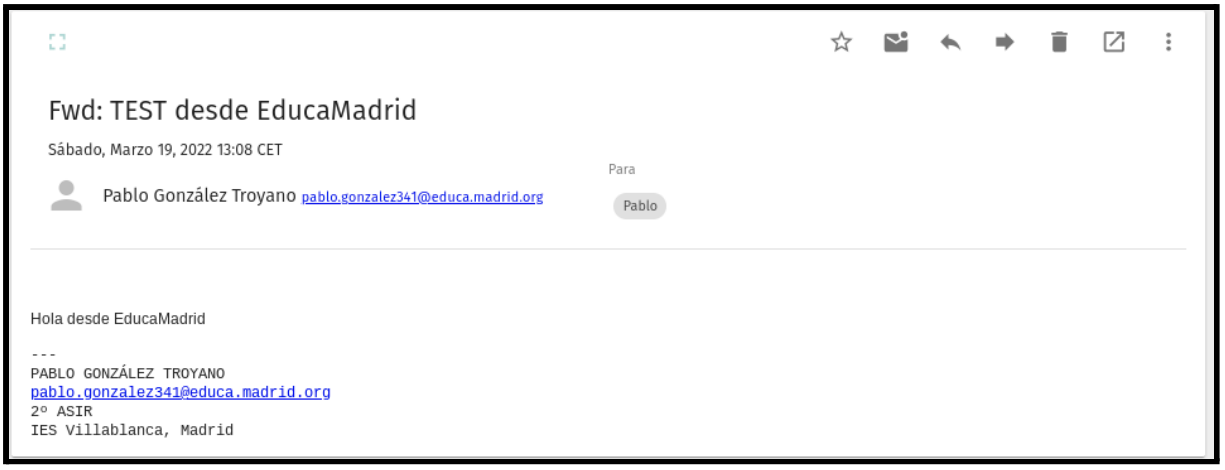
5.2.3.2. Con destino servidor de correo

También probaremos el envío desde EducaMadrid hacia Mailcow:



[60] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/2-mail/cabeceras-1.eml.txt

El mensaje recibido en Mailcow desde EducaMadrid es el que se puede ver en la siguiente imagen. Los mensajes sí salen desde EducaMadrid, pero no llegan.



Las cabeceras de este correo electrónico se encuentran también disponibles en el repositorio de GitHub[61], aunque también se mostrarán aquí las secciones interesantes de estas. De forma similar a la muestra anterior, las direcciones de correo electrónico se han modificado para evitar el Spam.

Para empezar vemos una cosa nada buena, la dirección IP de origen y el User-Agent son visibles de forma sencilla, ni siquiera una triste codificación en base64. Que es cutre, pero por lo menos no están a simple vista.

```
X-Remote-Browser: Mozilla/5.0 (X11; CrOS x86_64 14469.41.0) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/99.0.4844.57 Safari/537.36
X-Originating-IP: [79.116.7.222]
```

Por lo demás, también es interesante la cabecera que añade nuestro propio filtro de Spam al mensaje antes de entregarlo:

```
X-Spamd-Result: default: False [-2.00 / 15.00];
       DWL_DNSWL_LOW(-1.00)[madrid.org:dkim];
       DMARC_POLICY_ALLOW(-0.50)[madrid.org,reject];
       R_SPF_ALLOW(-0.20)[+ip4:130.206.19.0/24:c];
       R_DKIM_ALLOW(-0.20)[educa.madrid.org:s=dkim_educamadrid];
       MIME_GOOD(-0.10)[multipart/alternative,text/plain];
       MX_GOOD(-0.01)[];
       XM_UA_NO_VERSION(0.01)[];
       BCC(0.00)[];
       HAS_XOIP(0.00)[];
```

[61] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/2-mail/cabeceras-2.eml.txt

```
FROM_HAS_DN(0.00)[];
TO_MATCH_ENVRCPT_ALL(0.00)[];
ARC_NA(0.00)[];
RCPT_COUNT_ONE(0.00)[1];
PREVIOUSLY_DELIVERED(0.00)[pablo[RED]*^^@^^carpet4you.site];
RCPT_MAILCOW_DOMAIN(0.00)[carpet4you.site];
RCVD_TLS_LAST(0.00)[];
ASN(0.00)[asn:766, ipnet:130.206.0.0/16, country:ES];
RCVD_COUNT_THREE(0.00)[3];
HAS_ORG_HEADER(0.00)[];
ARC_SIGNED(0.00)[carpet4you.site:s=glezcloud:i=1];
MID_RHS_MATCH_FROM(0.00)[];
TO_DN_ALL(0.00)[];
FROM_EQ_ENVFROM(0.00)[];
DKIM_TRACE(0.00)[educa.madrid.org:+];
MIME_TRACE(0.00)[0:+,1:+,2:~];
RWL_MAILSPIKE_VERYGOOD(0.00)[130.206.19.171:from]
```

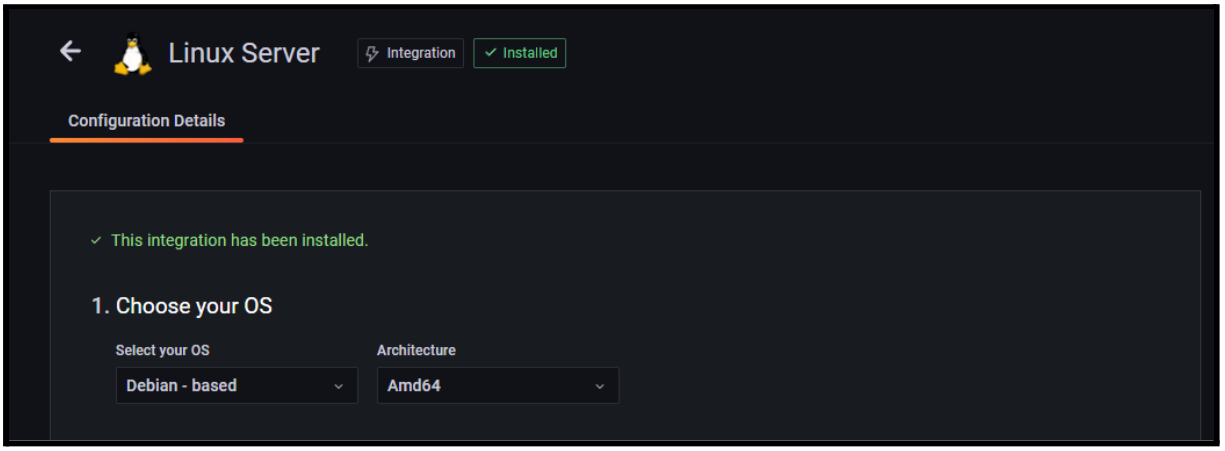
Por otro lado, también se ha respondido desde Google Workspace al mensaje de correo electrónico del apartado anterior. Ha llegado a la bandeja de entrada de forma correcta:

5.2.4. Monitorización del servidor

5.2.4.1. Instalación del agente de grafana

Antes de configurar el servidor se instaló el agente de monitorización de Grafana[62]. Este agente de monitorización recolecta información sobre CPU, memoria, red, etc. Además, envía los logs a Grafana de forma automática.

La instalación es sencilla. Desde el panel web de Grafana, nos situamos sobre *Integrations and Connections*. Entre todas las opciones, seleccionamos *Linux Server*. Escogemos nuestra arquitectura:



Ejecutamos el siguiente comando:

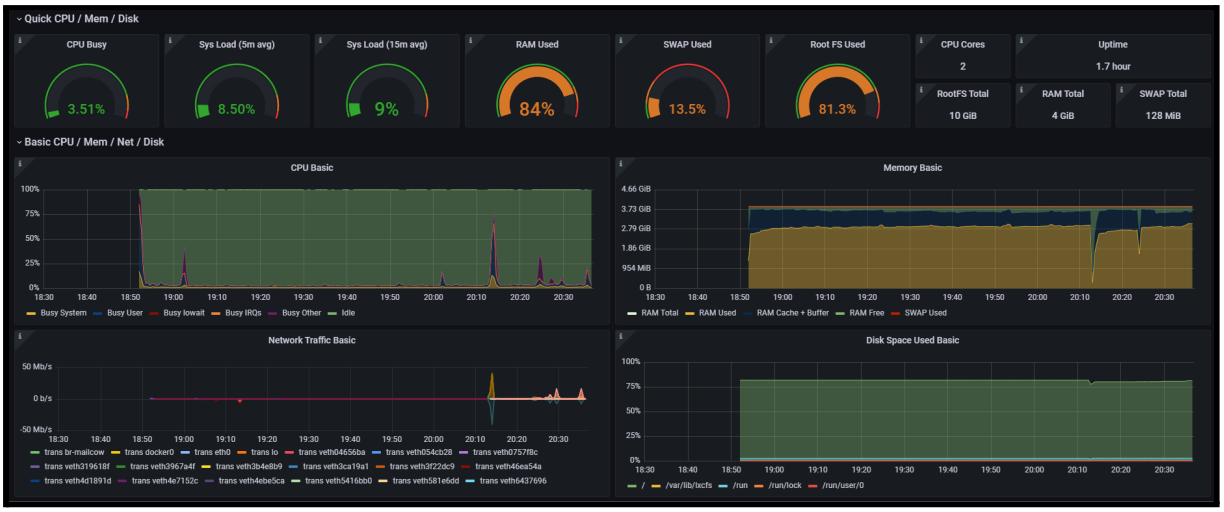
```
sudo ARCH=amd64 GCLOUD_STACK_ID="260642" GCLOUD_API_KEY="[ESTO ES SECRETO]"
GCLOUD_API_URL="https://integrations-api-eu-west.grafana.net" /bin/sh -c "$(curl
-fsSL
https://raw.githubusercontent.com/grafana/agent/release/production/grafanacloud-
install.sh)"
```

Y ya estaría enviando estadísticas a Grafana. Así de simple. Sin intención de realizar ninguna publicidad, indicar que Grafana Cloud tiene una versión gratuita[63] muy interesante, que nos permite obtener bastante visibilidad sobre nuestro entorno. Ofrece, sin coste 50 GB de Logs y 50 GB de trazas, 10 dashboards, 10.000 series de datos activas y más.

[62] https://grafana.com/oss/prometheus/exporters/node-exporter/
[63] https://grafana.com/pricing/

5.2.4.2. Monitorización - Métricas

De un vistazo, vemos el estado del servidor:



Los picos en el uso de CPU (gráfico temporal superior izquierdo), en el uso de la memoria RAM (gráfico temporal superior derecho), así como en la red (gráfico temporal inferior izquierdo) son debidos a pruebas de estrés que se han realizado enviando multitud de correos electrónicos, con archivos adjuntos pesados.

El servidor ha podido realizar sin problemas su cometido, sin experimentar degradaciones del servicio.

5.2.4.3. Monitorización - *Logs*

De forma automática, sin prácticamente ninguna configuración (se ha tenido que cambiar el hostname en el archivo `/etc/grafana-agent.yaml`) los registros del propio servidor son enviados a Grafana. Pero los registros realmente importantes son los que generan los contenedores. Debemos enviar estos registros también. Para hacerlo debemos, de forma previa a cualquier acción, instalar el plugin de registro de grafana:

```
docker plugin install grafana/loki-docker-driver:latest --alias loki
--grant-all-permissions
```

Podemos comprobar su instalación mediante el siguiente comando:

```
docker plugin ls
```

```
## La salida será similar a la siguiente:
ID             NAME          DESCRIPTION           ENABLED
bf9ee87bd37d   loki:latest   Loki Logging Driver   true
```

Ahora debemos editar nuestro docker-compose para añadir a cada contenedor el siguiente fragmento:

```
    logging:
      driver: loki
      options:
        loki-url: https://[user]:[PW]@[URL]/api/prom/push
```

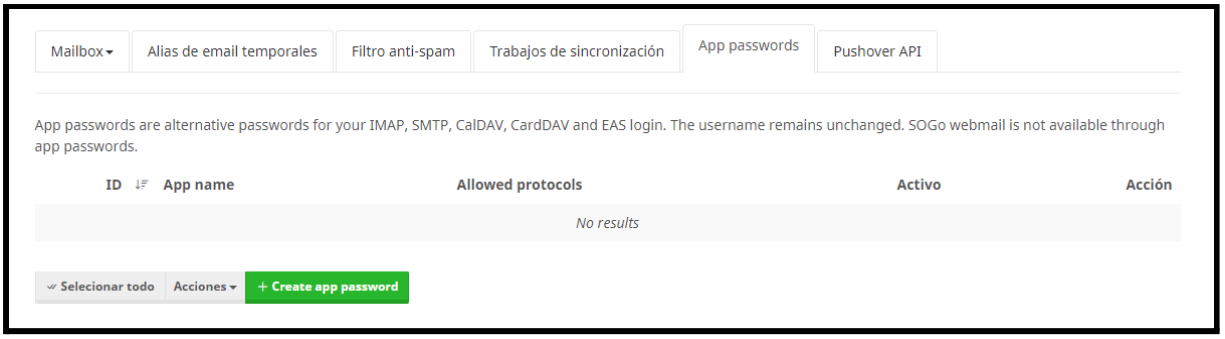
Puesto que había que hacerlo en cualquier caso para que los cambios se aplicaran, se ha comprobado como los logs de los reinicios se han volcado a Grafana en el histograma:

5.2.5. Conexión de un cliente IMAP

Iniciaremos sesión con nuestras credenciales (pablo^^@^^carpet4you.site) en la página principal de Mailcow (no en el webmail).

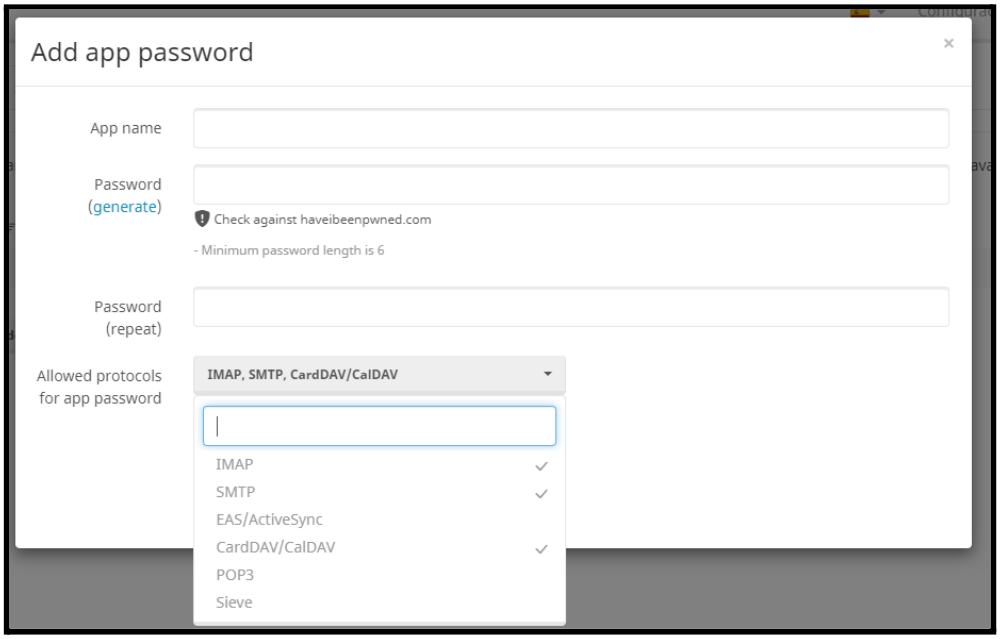
Este sistema tiene, entre sus muchas funciones, la posibilidad de crear contraseñas de aplicación específicas. De esta forma, cada cliente de correo electrónico puede tener una contraseña distinta, para prevenir problemas en caso de que una de estas contraseñas se vea comprometida.

Desde esta interfaz de gestión, en la parte superior nos dirigimos a la pestaña *App Passwords*. Una vez aquí, seleccionamos *Create app passwords*.



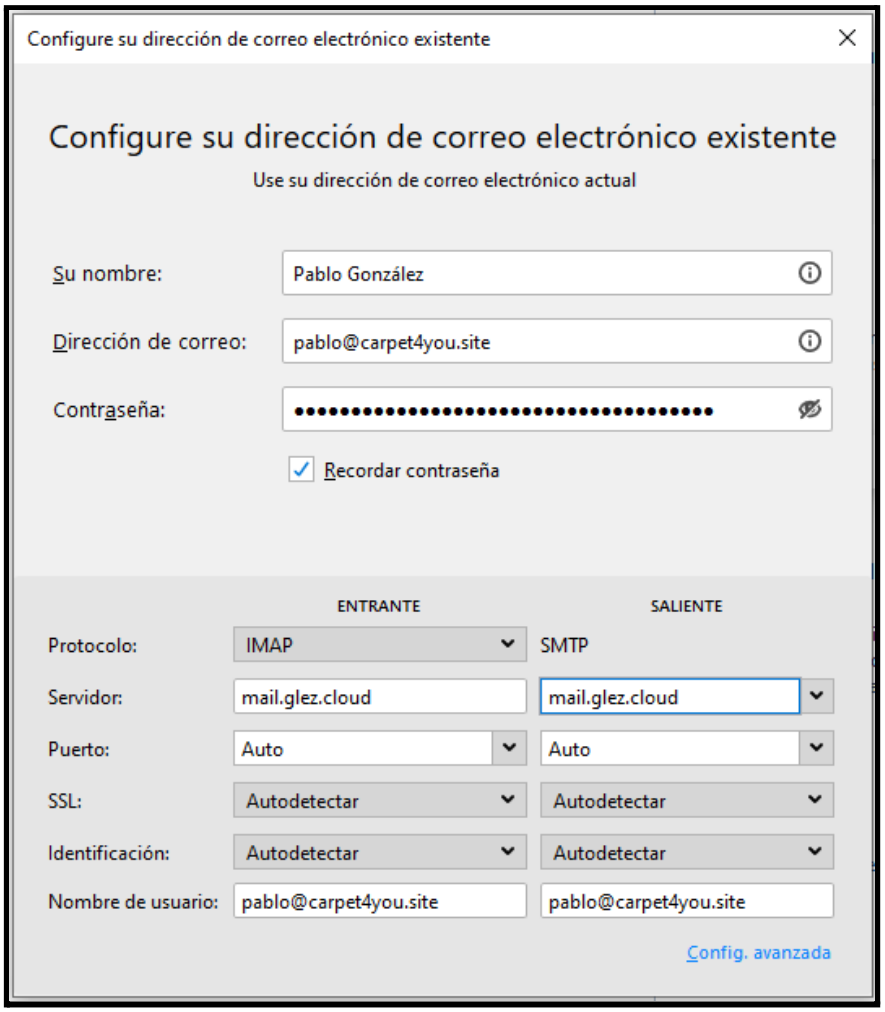
Al hacer clic en este botón nos aparecerá un cuadro de diálogo en el que debemos configurar una serie de opciones. Son las siguientes:

- Nombre de la aplicación. Realmente es un alias, pero nos permitirá identificar de forma sencilla la contraseña de aplicación generada.
- Contraseña. Será la contraseña en sí, lo más recomendable es que sea el propio sistema el que la genere de forma automática y aleatoria.
- Protocolos permitidos para la contraseña de aplicación. En nuestro caso, indicamos IMAP, SMTP y CardDAV/CalDAV.
- Activo. Aunque no lo veamos, debajo de la lista desplegable hay una casilla en la que podemos marcar si la contraseña de aplicación será válida.

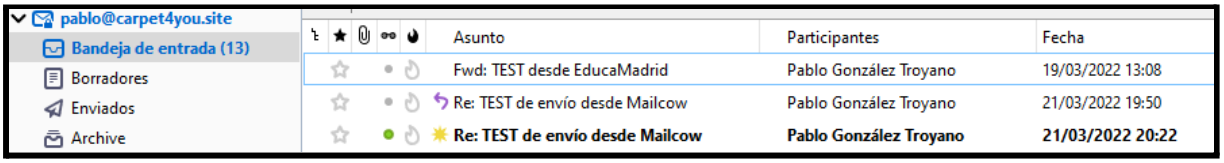


En nuestro caso, indicaremos *Thunderbird* en el nombre y seleccionaremos que el propio sistema genere la contraseña. Esta ha quedado registrada en el anexo III de este documento.

Ahora en el cliente Mozilla Thunderbird, indicamos los ajustes de SMTP e IMAP necesarios para realizar la conexión:



Veremos nuestro correo electrónico en este cliente:



También vamos a probar a enviar un correo electrónico desde el cliente de correo Mozilla Thunderbird para comprobar el correcto funcionamiento.

El mensaje ha llegado correctamente y en cuestión de pocos segundos:

La cabecera completa, con las ya mencionadas anteriormente modificaciones para evitar el spam ha sido subida al repositorio de GitHub[64] y se encuentra en el anexo correspondiente (Apartado 7.6.2, Anexo VI).

De forma adicional, se adjunta a continuación en este mismo documento el resumen que Google proporciona en lo relativo a identificadores, seguridad y cifrado:

ID de mensaje	<3a043d9b-d083-071f-23d0-5fff41a5b0f4@carpet4you.site>
Creado a las:	22 de marzo de 2022, 20:10 (entregado en -1 segundos)
De:	Pablo González <pablo@carpet4you.site>
Para:	[REDACTED]@gonzaleztroyano.es
Asunto:	Prueba de envío desde Thunderbird
SPF:	PASS con la IP 93.189.91.9 Más información
DKIM:	'PASS' con el dominio carpet4you.site Más información

5.2.6. Uso de la API

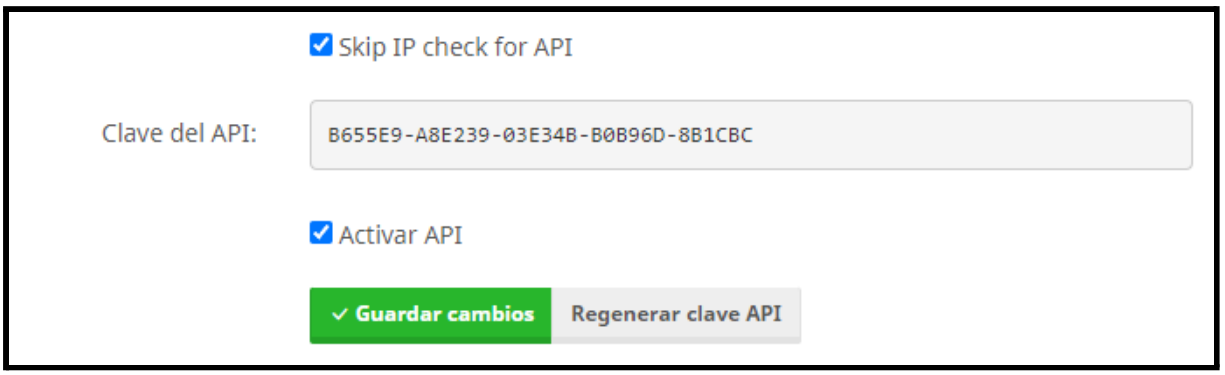
Usar las APIs nos permite acceder a la información de forma programática y sencilla, lo que nos permite automatizar ciertas tareas.

[64] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/2-mail/cabeceras-3.eml.txt

5.2.6.1. Generación de una contraseña API

Antes de realizar ninguna acción que implique APIs debemos generar una clave de API para autenticarnos contra el servidor. Lo podemos hacer accediendo con nuestro perfil de super administrador, desde la página principal:

Normalmente, indicaríamos unas IPs válidas para evitar que otros usuarios accedan. En nuestro caso, no lo activamos para simplificar la administración.



5.2.6.2. Pruebas con la API

Veamos algún ejemplo.

- Solicitar información sobre los dominios registrados

```
curl -X 'GET' \
  'https://mail.glez.cloud/api/v1/get/mailbox/all' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC'
```


```
[
  {
    "max_new_mailbox_quota": 7516192768,
    "def_new_mailbox_quota": 3221225472,
    "quota_used_in_domain": "3221225472",
    "bytes_total": "145019796",
    "msgs_total": "35",
    "mboxes_in_domain": 1,
    "mboxes_left": 9,
    "domain_name": "carpet4you.site",
    "description": "carpet4you.site",
    "max_num_aliases_for_domain": 400,
    "max_num_mboxes_for_domain": 10,
    "def_quota_for_mbox": 3221225472,
    "max_quota_for_mbox": 10737418240,
    "max_quota_for_domain": 10737418240,
    "relayhost": "0",
    "backupmx": 0,
    "backupmx_int": 0,
    "gal": 1,
```

```
    "gal_int": 1,
    "rl": false,
    "active": 1,
    "active_int": 1,
    "relay_all_recipients": 0,
    "relay_all_recipients_int": 0,
    "relay_unknown_only": 0,
    "relay_unknown_only_int": 0,
    "aliases_in_domain": 0,
    "aliases_left": 400,
    "domain_admins": "-"
  }
]
```


- Solicitar información sobre las cuentas de correo del sistema

```
curl -X 'GET' \
  'https://mail.glez.cloud/api/v1/get/domain/all' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC'
```


```
[
  {
    "username": "pablo@carpet4you.site",
    "active": 1,
    "active_int": 1,
    "domain": "carpet4you.site",
    "relayhost": null,
    "name": "Pablo González",
    "local_part": "pablo",
    "quota": 3221225472,
    "messages": 35,
    "attributes": {
      "force_pw_update": "0",
      "tls_enforce_in": "0",
      "tls_enforce_out": "0",
      "sogo_access": "1",
      "imap_access": "1",
      "pop3_access": "1",
      "smtp_access": "1",
      "sieve_access": "1",
      "relayhost": "0",
      "passwd_update": "2022-03-18 23:19:10",
      "mailbox_format": "maildir:",
      "quarantine_notification": "hourly",
      "quarantine_category": "reject"
    },
    "quota_used": 145019796,
    "percent_in_use": 5,
    "percent_class": "success",
    "last_imap_login": 1647976230,
    "last_smtp_login": 1647976221,
    "last_pop3_login": 0,
    "max_new_quota": 10737418240,
    "spam_aliases": 0,
    "pushover_active": 0,
    "rl": false,
    "rl_scope": "domain",
    "is_relayed": 0
  }
]
```

- Añadir una nueva cuenta de correo electrónico

```
curl -X 'POST' \
  'https://mail.glez.cloud/api/v1/add/mailbox' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC' \
  -H 'Content-Type: application/json' \
  -d '{
  "active": "1",
  "domain": "carpet4you.site",
  "local_part": "laura",
  "name": "Laura Machado",
  "password": "Temporal1234*",
  "password2": "Temporal1234*",
  "quota": "3072",
  "force_pw_update": "0",
  "tls_enforce_in": "1",
  "tls_enforce_out": "1"
}'
```


```
[
  {
    "type": "success",
    "log": [
      "mailbox",
      "add",
      "mailbox",
      {
        "active": "1",
        "domain": "carpet4you.site",
        "local_part": "laura",
        "name": "Laura Machado",
        "password": "*",
        "password2": "*",
        "quota": "3072",
        "force_pw_update": "0",
        "tls_enforce_in": "1",
        "tls_enforce_out": "1"
      },
      null
    ],
    "msg": [
      "mailbox_added",
      "laura@carpet4you.site"
    ]
  }
]
```


Se ha comprobado como Laura puede acceder sin problemas en el webmail

- Añadir dominio a Mailcow

```
curl -X 'POST' \
  'https://mail.glez.cloud/api/v1/add/domain' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC' \
  -H 'Content-Type: application/json' \
  -d '{
  "active": "1",
  "aliases": "400",
  "backupmx": "0",
  "defquota": "3072",
  "description": "Dominio interno súper importante ",
  "domain": "glez-cloud.tech",
  "mailboxes": "10",
  "maxquota": "10240",
  "quota": "10240",
  "relay_all_recipients": "0",
  "rl_frame": "s",
  "rl_value": "10",
  "restart_sogo": "10"
}'
```


```
[  {    "type": "success",
    "log": [
      "ratelimit",
      "edit",
      "domain",
      {
        "rl_value": "10",
        "rl_frame": "s",
        "object": "glez-cloud.tech"
      }
    ],
    "msg": [
      "rl_saved",
      "glez-cloud.tech"
    ]
  },
  {
    "type": "success",
    "log": [
      "mailbox",
      "add",
      "domain",
      {
        "active": "1",
        "aliases": "400",
        "backupmx": "0",
        "defquota": "3072",
        "description": "Dominio interno súper importante ",
        "domain": "glez-cloud.tech",
        "mailboxes": "10",
        "maxquota": "10240",
        "quota": "10240",
        "relay_all_recipients": "0",
        "rl_frame": "s",
        "rl_value": "10",
        "restart_sogo": "10"
      },
      null
    ],
    "msg": [
      "domain_added",
      "glez-cloud.tech"    ]  }]
```

Si ahora navegamos a través de la web GUI a la sección de dominios, veremos añadido el dominio mediante la API:

- Generemos la firma DKIM

```
curl -X 'POST' \
  'https://mail.glez.cloud/api/v1/add/dkim' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC' \
  -H 'Content-Type: application/json' \
  -d '{
  "dkim_selector": "glez-cloud",
  "domains": "glez-cloud.tech",
  "key_size": "1024"
}'
```

```
[
  {
    "type": "success",
    "log": [
      "dkim",
      "add",
      {
        "dkim_selector": "glez-cloud",
        "domains": "glez-cloud.tech",
        "key_size": "1024"
      }
    ],
    "msg": [
      "dkim_added",
      "glez-cloud.tech"
    ]
  }
]
```

- Obtengamos la firma DKIM generada en el paso anterior:

```
curl -X 'GET' \
  'https://mail.glez.cloud/api/v1/get/dkim/glez-cloud.tech' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC'
```


```
{
  "pubkey":
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3oPyRCoo2px6ZMXT1+2RF0MDYdLT5QA7rfRYxw9Wc4rWgvKy
AI2nqQ1/bic+hxy4AAQio+1kXcDrsn3sujrxtdYS0RnuyM2BftWd38SxE/27+eYMmVQ5AToTE1f8N/Ud8EJSdu3b
l8J6Id5LByD1EJT9ud0NACyrB0qwoiFvoaQIDAQAB",
  "length": "1024",
  "dkim_txt":
"v=DKIM1;k=rsa;t=s;s=email;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3oPyRCoo2px6ZMXT1+2R
F0MDYdLT5QA7rfRYxw9Wc4rWgvKyAI2nqQ1/bic+hxy4AAQio+1kXcDrsn3sujrxtdYS0RnuyM2BftWd38SxE/27
+eYMmVQ5AToTE1f8N/Ud8EJSdu3bl8J6Id5LByD1EJT9ud0NACyrB0qwoiFvoaQIDAQAB",
  "dkim_selector": "glez-cloud",
  "privkey": ""
}
```


Como vemos, el resultado no es muy “amigable” para utilizarlo en scripting. Por ello, vamos a pasarlo al comando `jq`, para que lo trate:

```
curl -X 'GET' \
  'https://mail.glez.cloud/api/v1/get/dkim/glez-cloud.tech' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC' | jq .dkim_txt
```


```
"v=DKIM1;k=rsa;t=s;s=email;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3oPyRCoo2px6ZMXT1+2R
F0MDYdLT5QA7rfRYxw9Wc4rWgvKyAI2nqQ1/bic+hxy4AAQio+1kXcDrsn3sujrxtdYS0RnuyM2BftWd38SxE/27
+eYMmVQ5AToTE1f8N/Ud8EJSdu3bl8J6Id5LByD1EJT9ud0NACyrB0qwoiFvoaQIDAQAB"
```

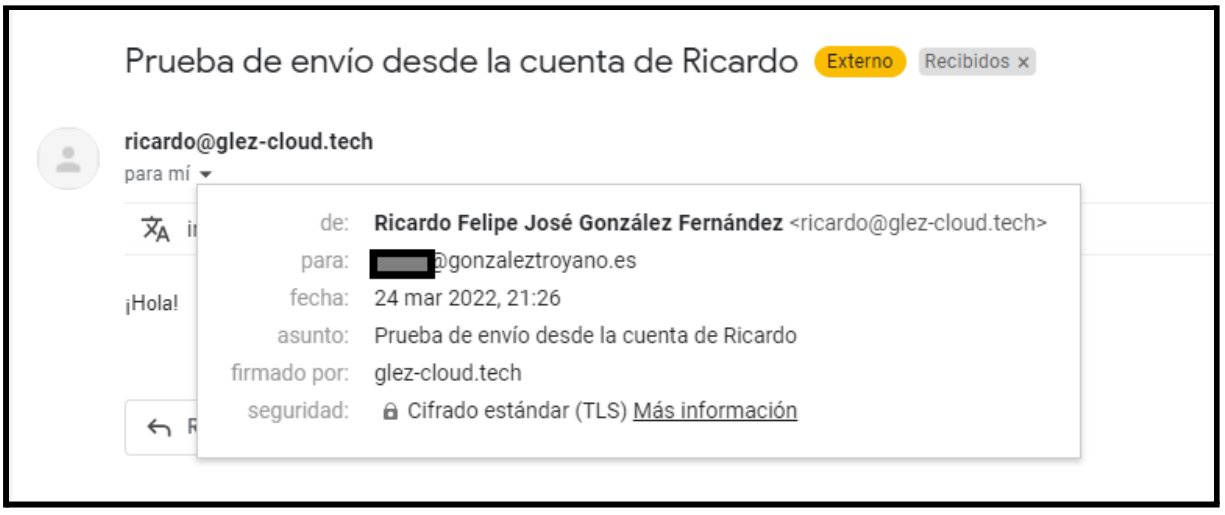

- Por último, creemos una cuenta de correo en este dominio:

```
curl -X 'POST' \
  'https://mail.glez.cloud/api/v1/add/mailbox' \
  -H 'accept: application/json' \
  -H 'X-API-Key: B655E9-A8E239-03E34B-B0B96D-8B1CBC' \
  -H 'Content-Type: application/json' \
  -d '{
  "active": "1",
  "domain": "glez-cloud.tech",
  "local_part": "ricardo",
  "name": "Ricardo Felipe José González Fernández",
  "password": "Temporal1234*",
  "password2": "Temporal1234*",
  "quota": "3072",
  "force_pw_update": "1",
```

```
  "tls_enforce_in": "1",
  "tls_enforce_out": "1"
}
```

```
[
  {
    "type": "success",
    "log": [
      "mailbox",
      "add",
      "mailbox",
      {
        "active": "1",
        "domain": "glez-cloud.tech",
        "local_part": "ricardo",
        "name": "Ricardo Felipe José González Fernández",
        "password": "*",
        "password2": "*",
        "quota": "3072",
        "force_pw_update": "1",
        "tls_enforce_in": "1",
        "tls_enforce_out": "1"
      },
      null
    ],
    "msg": [
      "mailbox_added",
      "ricardo@glez-cloud.tech"
    ]
  }
]
```


Se ha iniciado sesión en la cuenta de Ricardo y se ha enviado un correo electrónico de prueba. Ha llegado correctamente:



Se ha adjuntado en el anexo VI (Apartado 7.6.2) de este documento y se ha subido al repositorio disponible en GitHub[65].

[65] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/2-mail/cabeceras-4.eml.txt

5.3. Servicio web 🌐

El siguiente de los servicios que configuraremos de forma manual es servicio web/HTTP.

En esencia se ofrecerá a los clientes:

- Un espacio donde subir archivos HTML estáticos, imágenes, etc.
- Una conexión SFTP para poder acceder a su sitio web y gestionar los ficheros de este, así como los registros de las aplicaciones.
- Un sitio web, funcionando en WordPress.
- Se valorará, a su vez, la generación de scripts para la instalación de los siguientes CMS[66]:
    - Prestashop
    - Moodle
    - Magento
    - Joomla
    - Drupal
    - Ghost

Se continuará y mejorará el trabajo realizado para el módulo de Implantación de Aplicaciones Web. En el primer trimestre se creó un script para una función similar. Está alojado en este repositorio de GitHub[67].

Lo que se realizará para esta sección del Proyecto Fin de Ciclo será mejorarlo y ampliarlo para añadir funcionalidades y que el código sea más óptimo.

5.3.1 Rama de trabajo

Puesto que se venía utilizando GitHub como control de versiones, se ha creado una rama, en inglés *branch*, llamada updates-pfc[68] para no saturar la rama principal de

---

[66] Un sistema de gestión de contenidos o CMS (del inglés content management system) es un programa informático que permite crear un entorno de trabajo para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás usuarios.
Fuente: https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos

[67] https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/tree/main

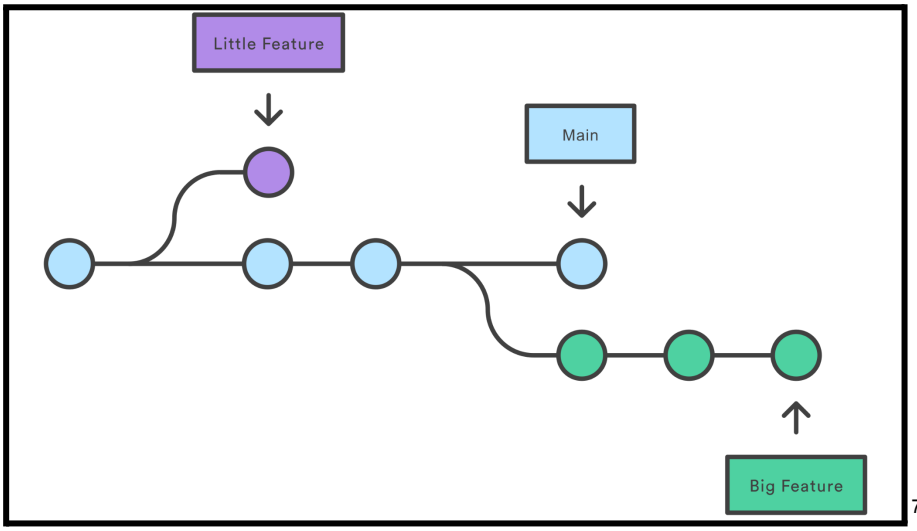
[68] https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/tree/updates-pfc

desarrollo. Esto es una buena práctica recomendada por todos los expertos en este tipo de sistemas. Para obtener más información sobre las ramas de código en los sistemas de control de versiones se puede acceder a este enlace[69] del proveedor Atlassian, que ofrece una alternativa comercial a GitHub, BitBucket. Ambas usan Git "por debajo" para la gestión.

En esta propia página podemos localizar la siguiente definición:

> *Una rama representa una línea independiente de desarrollo. Las ramas sirven como una abstracción de los procesos de cambio, preparación y confirmación. Puedes concebirlas como una forma de solicitar un nuevo directorio de trabajo, un nuevo entorno de ensayo o un nuevo historial de proyecto. Las nuevas confirmaciones se registran en el historial de la rama actual, lo que crea una bifurcación en el historial del proyecto.*

Se denomina "ramas" debido a la forma de representarlas:



[70]

Para ver los cambios entre dos ramas de un control de versiones basado en Git podemos usar:

```
git diff main..updates-pfc
```

[69] https://www.atlassian.com/es/git/tutorials/using-branches
[70] Imagen obtenida de la página nota al pie de página número 69. Bajo Licencia CC BY 2.5 AU

Sin embargo, el comando anterior muestra quizá demasiada información. Si únicamente queremos listar los *commits* en la nueva rama podemos usar:

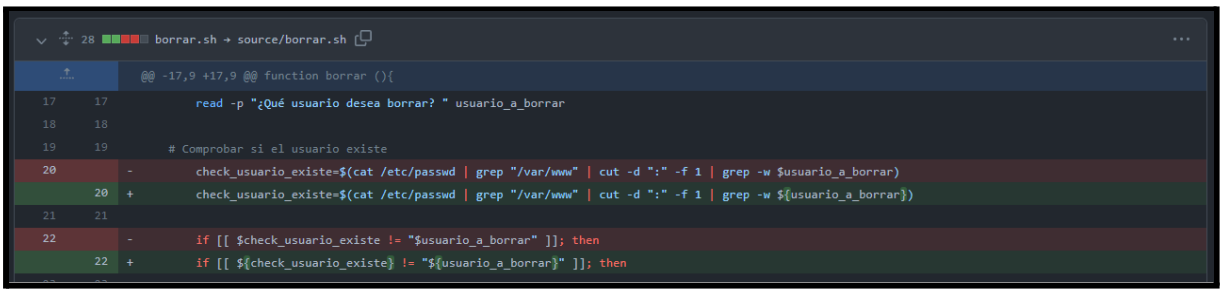
```
git log --oneline --graph --decorate --abbrev-commit main..updates-pfc
```

La salida de este comando es la siguiente:

```
Pablo@WIN-PABLO MINGW64 /d/github/ASIR2-IAW-SCRIPT (updates-pfc)
$ git log --oneline --graph --decorate --abbrev-commit main..updates-pfc
* def9070 (HEAD -> updates-pfc, origin/updates-pfc) Add CF Cleaner and minor changes
* f207464 Add curly braces to variables Using \$([^{\n{}() ".;\\\/:',-]{1,}) as regex and ${$1} as groupselector
* b3a9370 Fix Avoid WordPress Download on every install #36
* 61bc0e4 Install Prestashop Prev Fixes: Enviar correo con la contraseña cambiada. #39 Prev Fixes: Poder crear solo un WP #6
* c9ce205 Antes de prestashop
* e84ce48 Cambios variados
* 8bfb1a8 Cambios pequeños No va el add_app.
* 5be8ca4 220405-1133
* 88a3090 Pequeños avances bb
* ed1cf43 Por si revienta 2
* f6ceb7d Por si revienta
* 37086b7 Cambios 3103-1
* d8f4880 Move to source dir & improvements
```

Desde la propia interfaz web de GitHub[71] también podemos consultar la información:

También podemos ver los cambios individuales para cada archivo:



[71] https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/compare/updates-pfc?expand=1

5.3.2. Organización del script

Usando el comando `tree` en el servidor web, cuyo resultado se encuentra a la derecha de este texto, podemos ver cómo se organizan los diferentes archivos que conforman la utilidad.

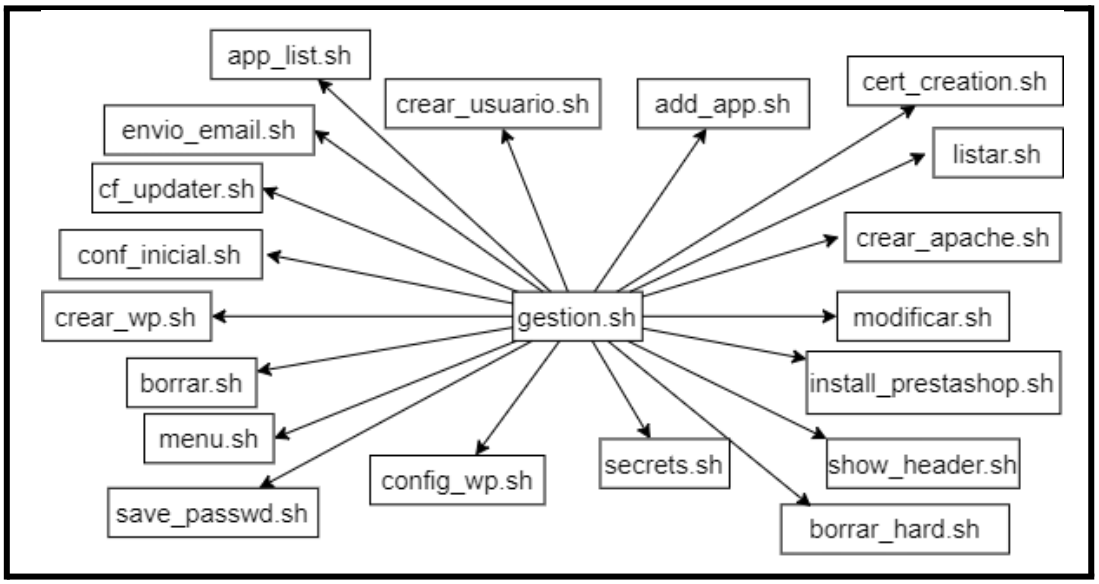
Puesto que la parte inicial del código ya estaba desarrollado en Bash Script, se ha continuado el desarrollo de la utilidad en este.

Está organizado en base a funciones, de forma que cada una únicamente se ejecuta cuando es necesario y una función va “llamando” a otras según sea necesario.

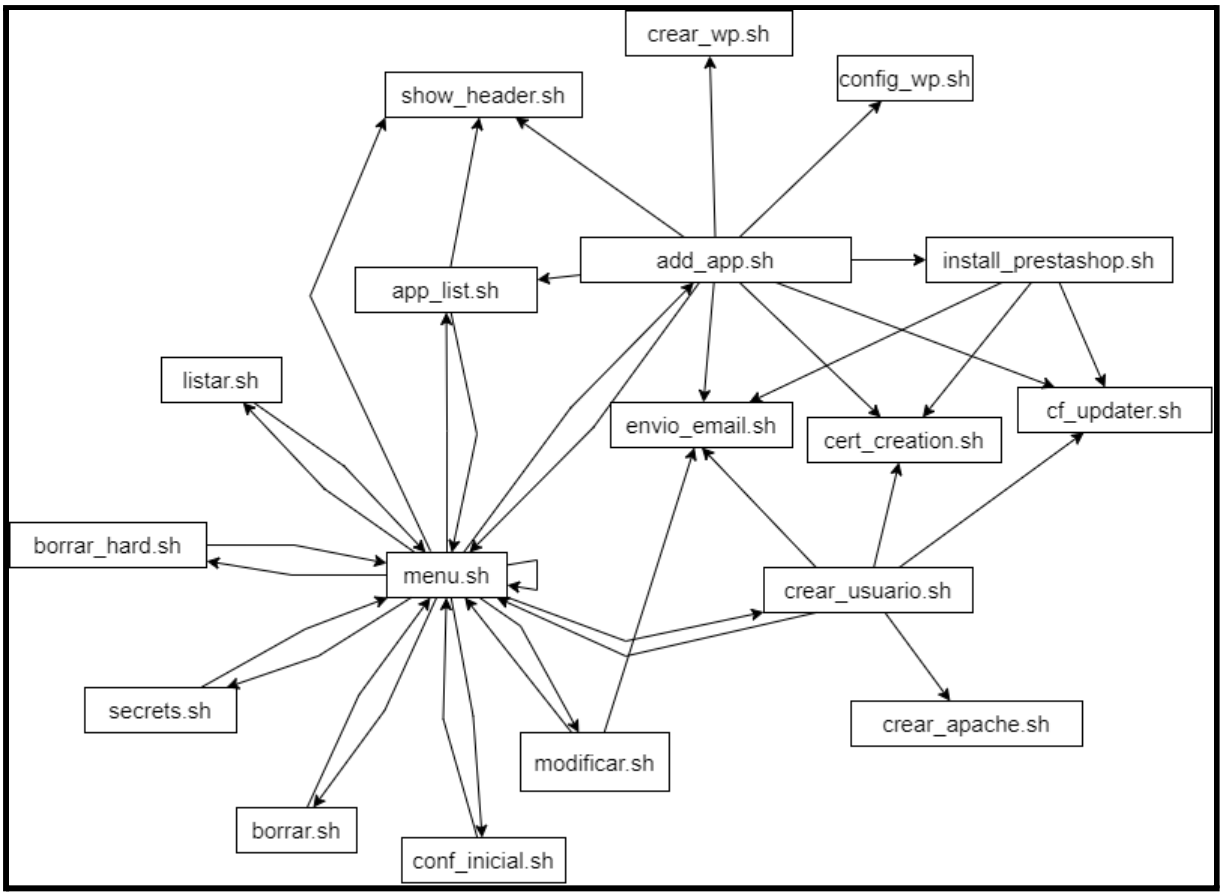
Organizarlo en base a funciones también ayuda a facilitar el desarrollo y el mantenimiento de la utilidad.

```
.
├── gestion.sh
└── source
    ├── add_app.sh
    ├── app_list.sh
    ├── borrar.sh
    ├── borrar_hard.sh
    ├── cert_creation.sh
    ├── cf_updater.sh
    ├── conf_inicial.sh
    ├── config_wp.sh
    ├── crear_apache.sh
    ├── crear_usuario.sh
    ├── crear_wp.sh
    ├── envio_email.sh
    ├── install_prestashop.sh
    ├── listar.sh
    ├── menu.sh
    ├── modificar.sh
    ├── save_passwd.sh
    ├── secrets.sh
    └── show_header.sh
```

El script gestion.sh es el principal. Es el encargado de cargar todas las funciones para usarlas posteriormente:



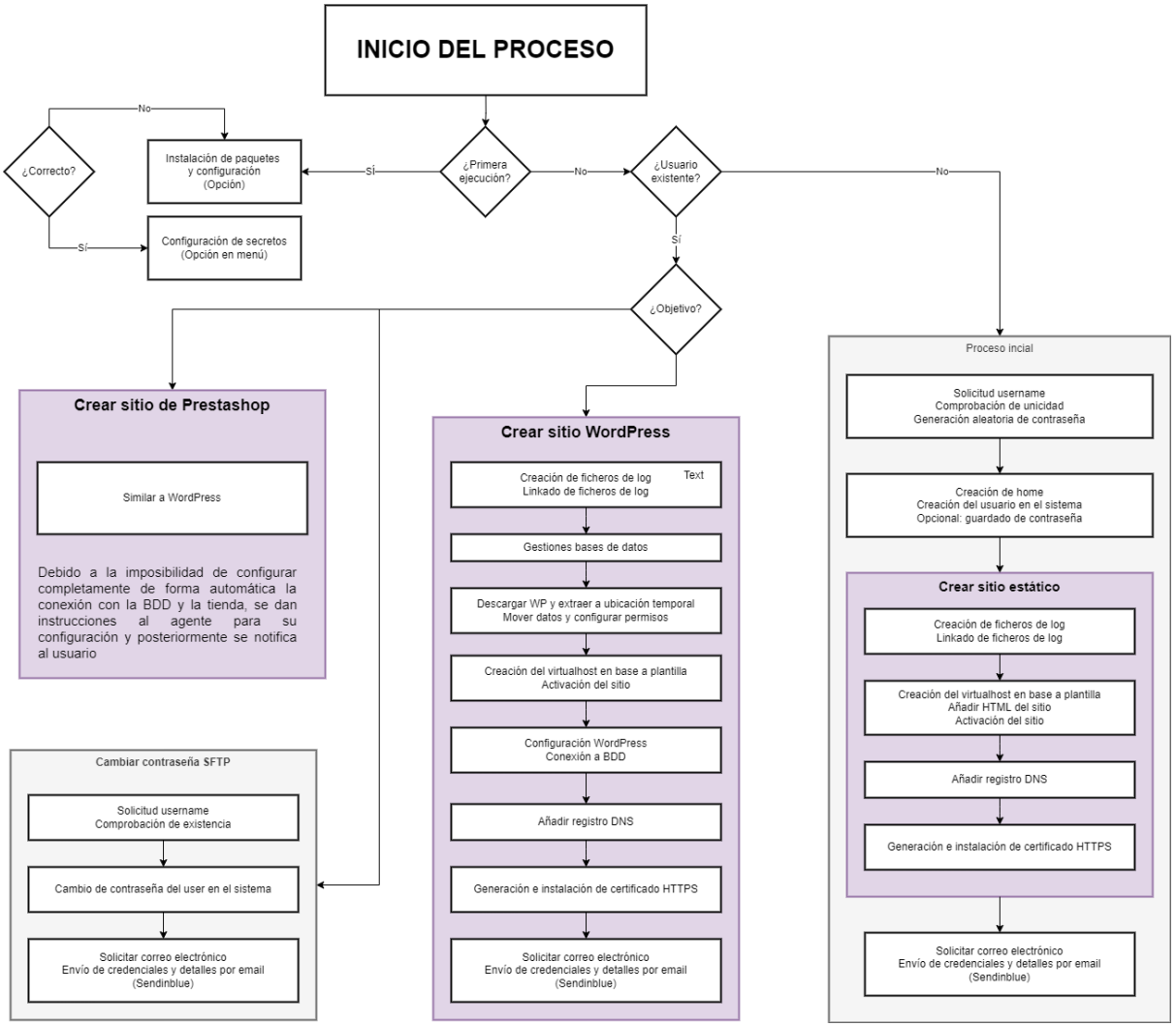
En tanto a las dependencias internas de las funciones:



A destacar en este gráfico de dependencias:

- Se ha generado de forma automática utilizando una declaración en formato texto/mermaid y el programa diagrams.net. Este enfoque permite realizar control de versiones también sobre esta parte.
- Las funciones que no retornan a *menu.sh* son “operativas”. Es decir, realizan funciones para otras. Es el caso, por ejemplo, de *cert_creation.sh*. No es una función principal, sino que es llamada por otras para completar una tarea (en el diagrama se puede ver como es llamada por *add_app.sh*, *crear_usuario.sh* e *install_prestashop*.sh).
- La función, sin tener en cuenta *menu.sh*, que más subfunciones llama es *add_app.sh*.
- La función más llamada es *envio_email.sh*, sin tener en cuanta menu.sh.

El ciclo de vida completo de los procesos se puede resumir en el siguiente diagrama:



5.3.3. Demostración de funcionamiento

Todo el código de estos scripts se encuentra, además del anteriormente citado repositorio, en el *Anexo VII: Códigos relativos al servicio web* de este documento.

5.3.3.1. Menú principal

Este menú en un primer momento es llamado por el script *gestion.sh* (que es el que se usará siempre para iniciar la utilidad). Permite seleccionar, de un listado definido, una opción. Muestra información sobre la licencia y el autor. Si la opción indicada no

se corresponde con ninguna de las opciones, vuelve a solicitar al usuario la selección de una.

5.3.3.2. Configuración inicial del servidor

Esta utilidad permite instalar los paquetes necesarios para que el servidor web funcione correctamente. También modifica y/o crea archivos de configuración.

No pide ninguna entrada al usuario, es bastante directo:

Por seguridad y facilitar el *troubleshooting*, guarda en el archivo `/tmp/conf_inicial.log` todos los logs relacionados con la instalación. En caso de que fallara, remitiría al usuario a este archivo.

Los paquetes que instala son: *apache2, php, libapache2-mod-php, libapache2-mod-php, php-mysql, php-cli, mariadb-server, mariadb-client, php-curl, php-gd, php-mbstring, php-xml, php-xmlrpc, php-soap, php-intl, php-zip, libapache2-mpm-itk, apt-utils, jq* y *certbot python3-certbot-apache*.

Activa el módulo “rewrite” de apache y reinicia el servicio para que se apliquen los cambios.

Crea una copia (por seguridad) del archivo de configuración del servidor SSH y lo sustituye por el siguiente, que descarga desde el repositorio de GitHub[72]:

```
PermitRootLogin no
LoginGraceTime 60
Subsystem sftp internal-sftp
PrintMotd no
SyslogFacility AUTH
LogLevel INFO
MaxAuthTries 2
MaxSessions 2
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM yes
#X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*
ClientAliveInterval 120
UseDNS no

Match User marcador
    ChrootDirectory %h
    ForceCommand internal-sftp -u 0027
    PasswordAuthentication yes
```

Estos cambios son ajustes de seguridad. Limitar sesiones e intentos de inicio de sesiones a 2, desactivar el inicio de sesión mediante contraseña y para el usuario root, definir el subsistema para el SFTP.

Además, se añade una sección muy importante, resaltada en azul cian en la configuración. Esta sección es lo que permite que los usuarios web del servidor sí puedan acceder mediante contraseñas, les encierra en su directorio *home* y fuerza a que únicamente puedan usar el servidor SFTP.

[72] https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/blob/main/templates%20and%20misc/sshd_config

5.3.3.3. Configuración de secretos

Puesto que durante la ejecución de las diferentes funciones son realizadas llamadas a APIs y se ha intentado que el script fuera útil para varios dominios, se han de configurar una serie de variables (“secretos”) antes de ejecutarlo.

Esta sección es también realmente sencilla, solicita al usuario una serie de datos. Almacena en variables “locales” (por denominarlas de alguna manera). Después, las guarda en el archivo que BASH utiliza para cargar la configuración de la sesión del usuario *~/.bashrc*. Además, se encarga de asegurarse de que serán exportadas en cada inicio de sesión del usuario root.

La interfaz de usuario se puede ver en la siguiente imagen. Nótese que los valores indicados no se corresponden con ningún tipo de clave o secreto real y su fin es meramente demostrativo.

5.3.3.4. Listar usuarios

Al seleccionar la opción 1 en el menú, nos pregunta si queremos filtrar algún usuario en concreto.

```
  9. Borrar usuario directamente (sin preguntar)

     Opción seleccionada: 1

¿Desea buscar algún nombre de usuario en concreto? [s/N]:
```

La opción por defecto es no aplicar ningún filtro. En caso de que sí queramos aplicar filtros, basta con pulsar “s” e *Intro* a continuación. En este caso, nos solicita introducir el término de búsqueda:

```
     Opción seleccionada: 1

¿Desea buscar algún nombre de usuario en concreto? [s/N]: s
Introduzca el término a buscar:
```

En este caso, como demostración, vamos a introducir “141”. Al pulsar *Intro*, vemos el resultado:

```
¿Desea buscar algún nombre de usuario en concreto? [s/N]: s
Introduzca el término a buscar: 141
Estos son los usuarios que coinciden con el término indicado:

ocitest141

 -- FIN DE LA LISTA --

Pulse cualquier tecla para volver al menú.
```

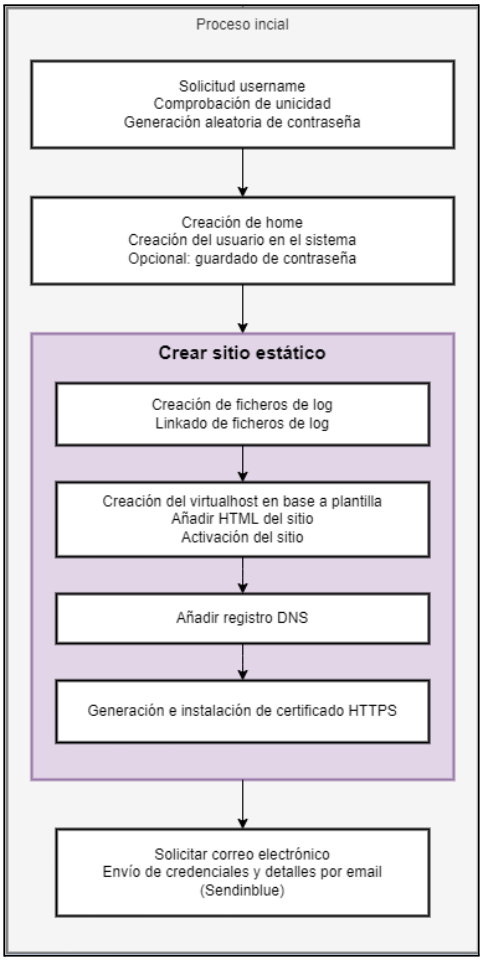
Pulsando cualquier tecla, volvemos al menú.

Si en la primera opción indicamos que no queremos filtrar (o lo que es lo mismo, pulsamos *Intro* directamente), nos listará todos los usuarios web del sistema:

```
¿Desea buscar algún nombre de usuario en concreto? [s/N]:
Usuarios del sistema web:

www-data
ocidev2
ocitest3
ocitest6
oci-test-5-3
oci-test-6-1
oci-test-6-3
oci-test-12-4
oci-test-12-5
ocitest12final
ocitest141
ocitest142
ocitest143

 -- FIN DE LA LISTA --

Pulse cualquier tecla para volver al menú.
```

5.3.3.5. Crear usuario nuevo: e-mail, certificados y SFTP

Este proceso, del cual se puede ver el esquema a la derecha de este texto, es el que se inicia cuando se crea un nuevo usuario en el sistema.



Se solicita el nombre de usuario al operador/a. Se comprueba que es único y se genera una contraseña aleatoria. A continuación, su directorio *home* es creado y es dado de alta en el sistema (utilizando la contraseña aleatoria y el directorio para la *home*). Este es el fragmento del código:

```
useradd -M -U --home
/var/www/${usuario_nuevo} --shell /bin/bash
${usuario_nuevo}
```

Para que el usuario pueda acceder a los *logs* de sus sitios a través de SFTP, se crean los archivos de log y se *linkan*, para que lo escrito en una ruta

también se escriba en la otra. Se realiza con este fragmento en el script:

```
                ln /var/log/apache2/${1}.${global_base_domain}.log
        /var/www/${1}/ficheros/logs/${1}.${global_base_domain}.log
```

En base a la siguiente plantilla, también disponible en el repositorio de GitHub[73]. Se crea el archivo de configuración de Virtualhost. Utilizando el comando *sed* se sustituyen los valores para aplicar la configuración del usuario con el que se está trabajando.

```
 <VirtualHost *:80>
    ServerName USER-TO-CHANGE.GLOBAL-BASE-DOMAIN
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/USER-TO-CHANGE/web
    ErrorLog /var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN.log
   CustomLog /var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-access.log combined
    AssignUserID USER-TO-CHANGE USER-TO-CHANGE
</VirtualHost>
```

Este es el fragmento del script encargado de esta función:

```
wget -qO /etc/apache2/sites-available/${1}.conf
https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/template
s%20and%20misc/virtualhost.txt

sed -i "s/USER-TO-CHANGE/${1}/g" "/etc/apache2/sites-available/${1}.conf"

sed -i "s/GLOBAL-BASE-DOMAIN/${global_base_domain}/g"
"/etc/apache2/sites-available/${1}.conf"
```

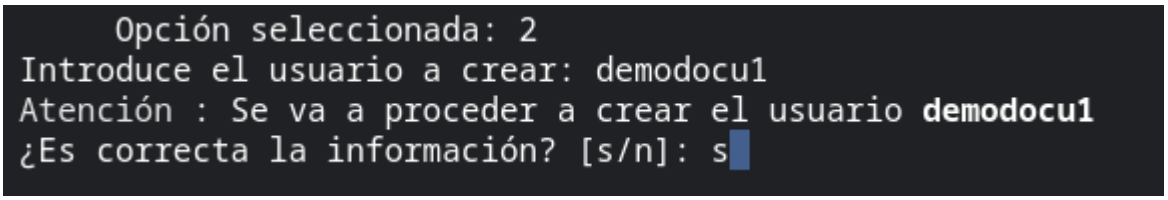
Acto seguido, se crea un index.html dando la bienvenida al usuario. Se configura el servicio SSH para permitir que el nuevo usuario se conecte mediante SFTP utilizando el marcador añadido al archivo de configuración del *demonio* SSH. Este es el fragmento que realiza la operación sobre la configuración de SSH:

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config_bak
touch /tmp/sshd_config
sed -r "s/^(Match User marcador.*$)/\1,${1}/" "/etc/ssh/sshd_config" >
          /tmp/sshd_config
mv /tmp/sshd_config /etc/ssh/sshd_config
```

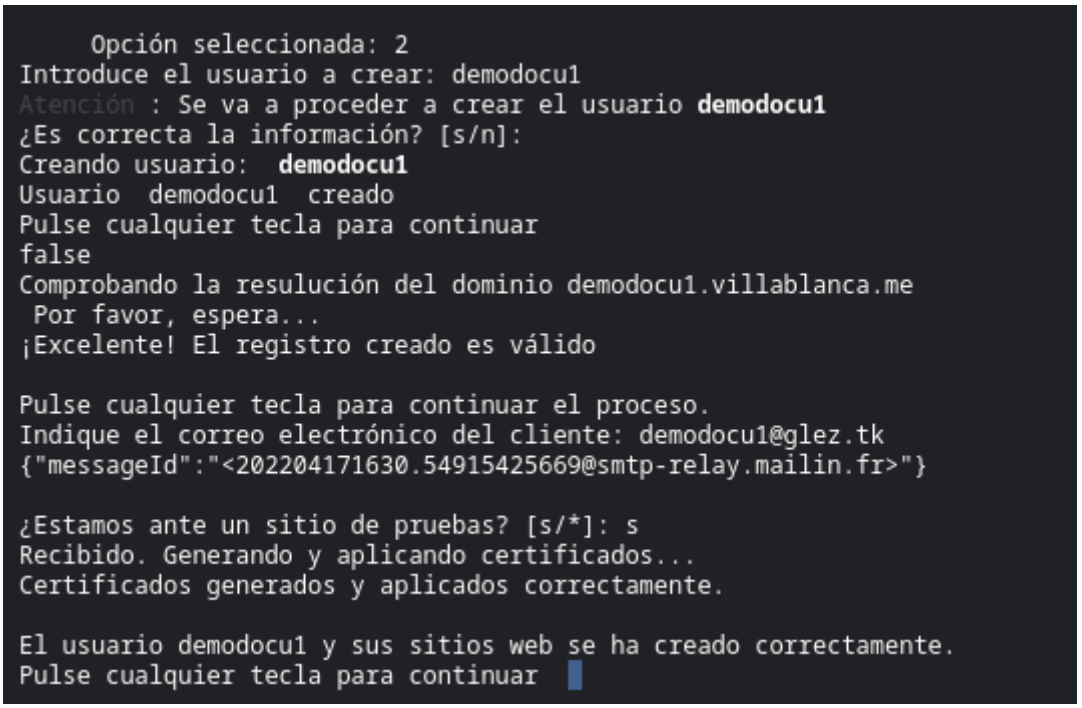
[73] https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/blob/main/templates%20and%20misc/virtualhost.txt

Después, se actualiza el registro DNS, se aplican los certificados TLS y se envía un mensaje de correo electrónico con los datos del sitio al usuario.

Veamos una demostración del proceso. En el menú, basta con seleccionar la opción 2. A continuación, nos solicitará introducir el identificador del usuario a crear. Internamente, el script comprueba que es único y de serlo, nos pide confirmación para el usuario a crear. El agente debe confirmar que la información es correcta para continuar el proceso.



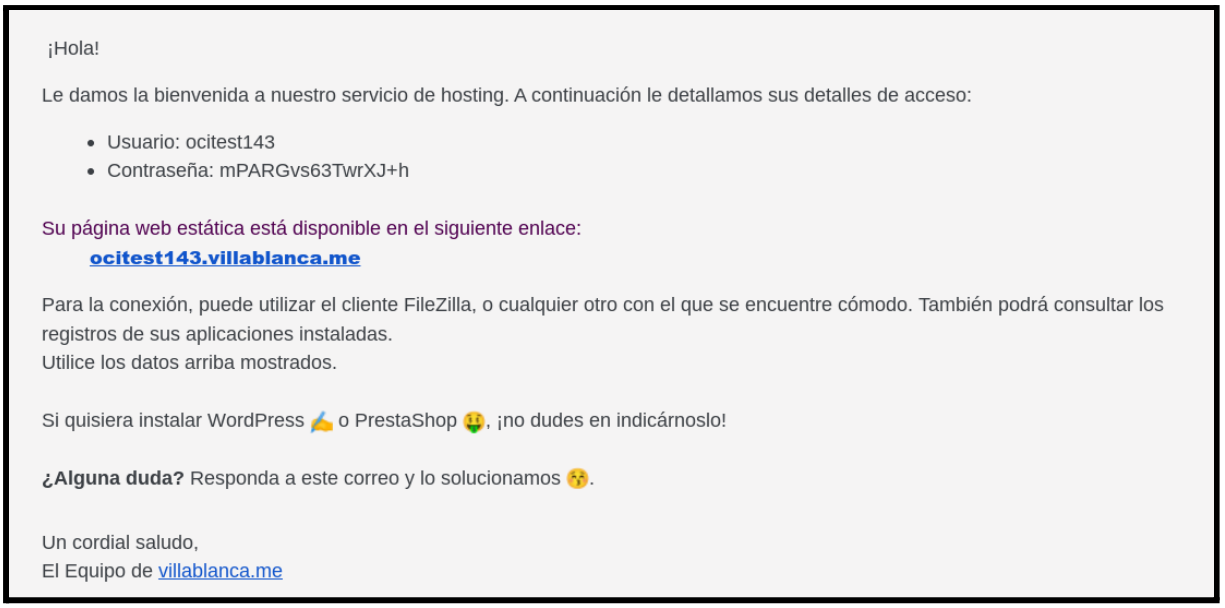
El proceso continúa de la forma que se puede ver en la siguiente captura de pantalla.



La impresión en formato PDF se encuentra en este enlace[74], además se adjunta a continuación una captura de pantalla con la información clave recibida.

[74] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/3-webservice-evidences/correo_new_user.pdf

El correo es enviado utilizando la API de Sendinblue.



¡Hola!

Le damos la bienvenida a nuestro servicio de hosting. A continuación le detallamos sus detalles de acceso:

- Usuario: ocitest143
- Contraseña: mPARGvs63TwrXJ+h

Su página web estática está disponible en el siguiente enlace:
ocitest143.villablanca.me

Para la conexión, puede utilizar el cliente FileZilla, o cualquier otro con el que se encuentre cómodo. También podrá consultar los registros de sus aplicaciones instaladas.
Utilice los datos arriba mostrados.

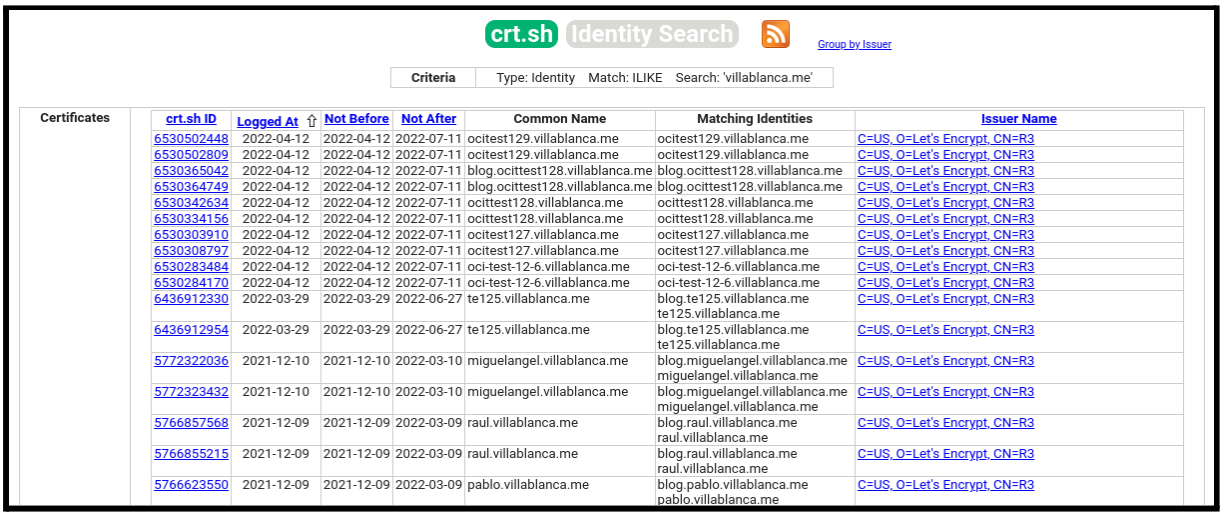
Si quisiera instalar WordPress ✍️ o PrestaShop 🤑, ¡no dudes en indicárnoslo!

¿Alguna duda? Responda a este correo y lo solucionamos 😗.

Un cordial saludo,
El Equipo de villablanca.me

Para los certificados, como se puede ver en una captura de pantalla anterior, se pregunta al operador si el sitio que se está configurando es un sitio de pruebas.

Esto es algo que no se utilizaría en producción, pero debido a las necesidades de desarrollo (muchas pruebas = muchos certificados emitidos) se ha añadido para evitar "saturar" el servidor de Let's Encrypt y, sobre todo evitar añadir demasiadas entradas a los logs de transparencia[75] de certificados, ya de por sí con bastantes entradas[76]:



crt.sh Identity Search — Group by Issuer

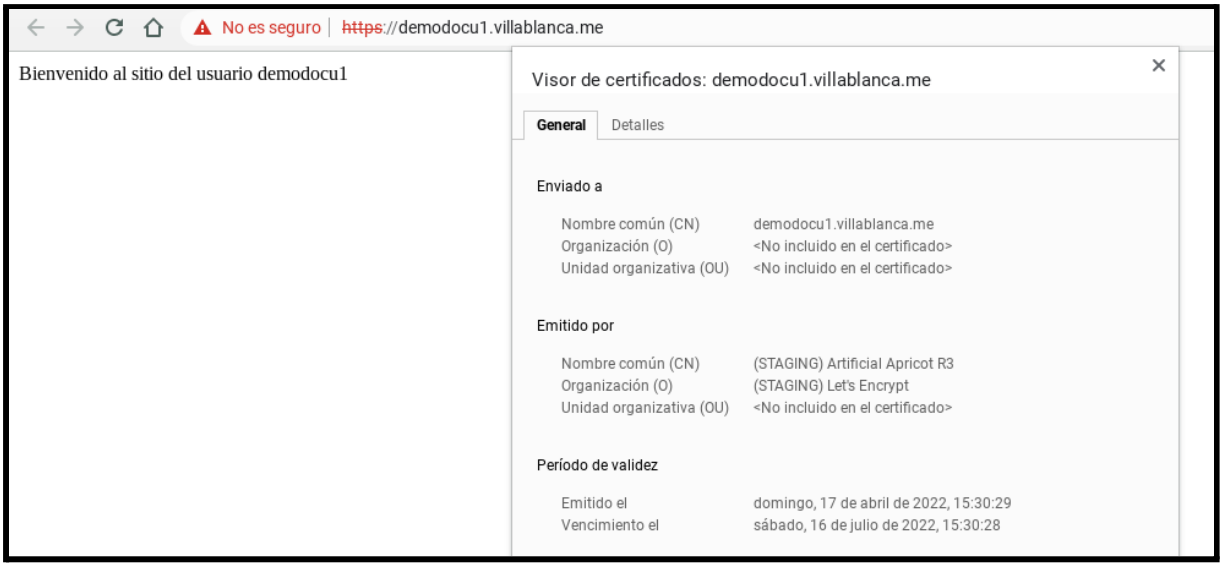
Criteria: Type: Identity Match: ILIKE Search: 'villablanca.me'

Certificates

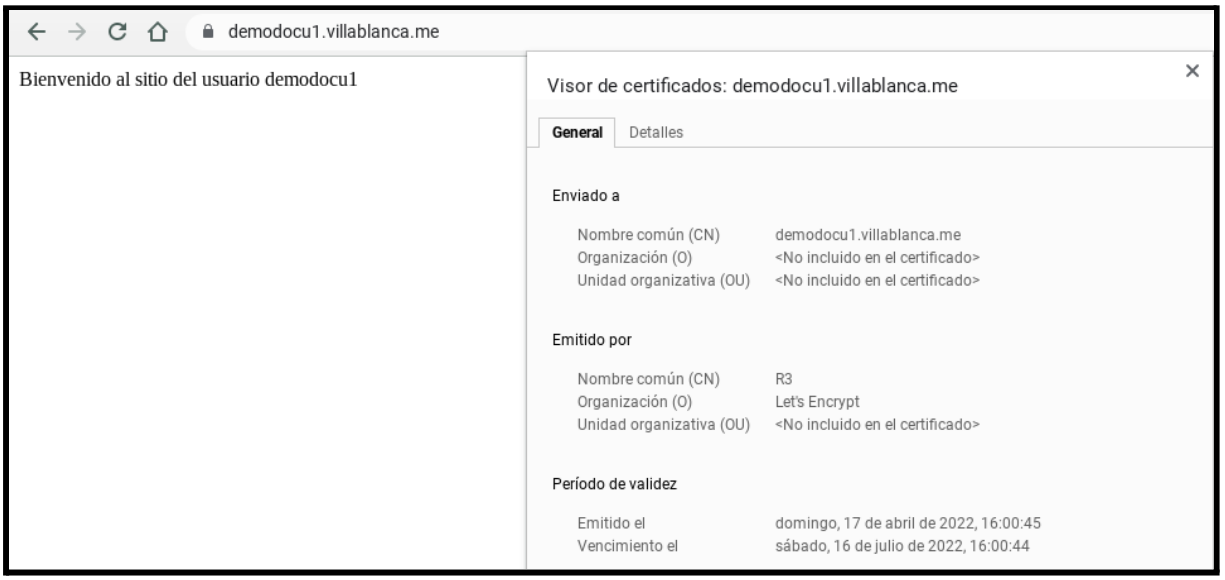
crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
6530502448	2022-04-12	2022-04-12	2022-07-11	ocitest129.villablanca.me	ocitest129.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530502809	2022-04-12	2022-04-12	2022-07-11	ocitest129.villablanca.me	ocitest129.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530365042	2022-04-12	2022-04-12	2022-07-11	blog.ocittest128.villablanca.me	blog.ocittest128.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530364749	2022-04-12	2022-04-12	2022-07-11	blog.ocittest128.villablanca.me	blog.ocittest128.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530342634	2022-04-12	2022-04-12	2022-07-11	ocittest128.villablanca.me	ocittest128.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530334156	2022-04-12	2022-04-12	2022-07-11	ocittest128.villablanca.me	ocittest128.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530303910	2022-04-12	2022-04-12	2022-07-11	ocitest127.villablanca.me	ocitest127.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530308797	2022-04-12	2022-04-12	2022-07-11	ocitest127.villablanca.me	ocitest127.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530283484	2022-04-12	2022-04-12	2022-07-11	oci-test-12-6.villablanca.me	oci-test-12-6.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6530284170	2022-04-12	2022-04-12	2022-07-11	oci-test-12-6.villablanca.me	oci-test-12-6.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6436912330	2022-03-29	2022-03-29	2022-06-27	te125.villablanca.me	blog.te125.villablanca.me te125.villablanca.me	C=US, O=Let's Encrypt, CN=R3
6436912954	2022-03-29	2022-03-29	2022-06-27	te125.villablanca.me	blog.te125.villablanca.me te125.villablanca.me	C=US, O=Let's Encrypt, CN=R3
5772322036	2021-12-10	2021-12-10	2022-03-10	miguelangel.villablanca.me	blog.miguelangel.villablanca.me miguelangel.villablanca.me	C=US, O=Let's Encrypt, CN=R3
5772323432	2021-12-10	2021-12-10	2022-03-10	miguelangel.villablanca.me	blog.miguelangel.villablanca.me miguelangel.villablanca.me	C=US, O=Let's Encrypt, CN=R3
5766857568	2021-12-09	2021-12-09	2022-03-09	raul.villablanca.me	blog.raul.villablanca.me raul.villablanca.me	C=US, O=Let's Encrypt, CN=R3
5766855215	2021-12-09	2021-12-09	2022-03-09	raul.villablanca.me	blog.raul.villablanca.me raul.villablanca.me	C=US, O=Let's Encrypt, CN=R3
5766623550	2021-12-09	2021-12-09	2022-03-09	pablo.villablanca.me	blog.pablo.villablanca.me pablo.villablanca.me	C=US, O=Let's Encrypt, CN=R3

[75] https://certificate.transparency.dev/
[76] https://crt.sh/?q=villablanca.me

Si en el selector se indica que sí estamos ante un sitio de prueba el certificado es emitido por la CA de Staging de Let's Encrypt[77], aquí vemos un ejemplo del certificado emitido por esta CA. Nótese que no es aceptado por los navegadores como confiable, pero técnicamente el proceso es equivalente:



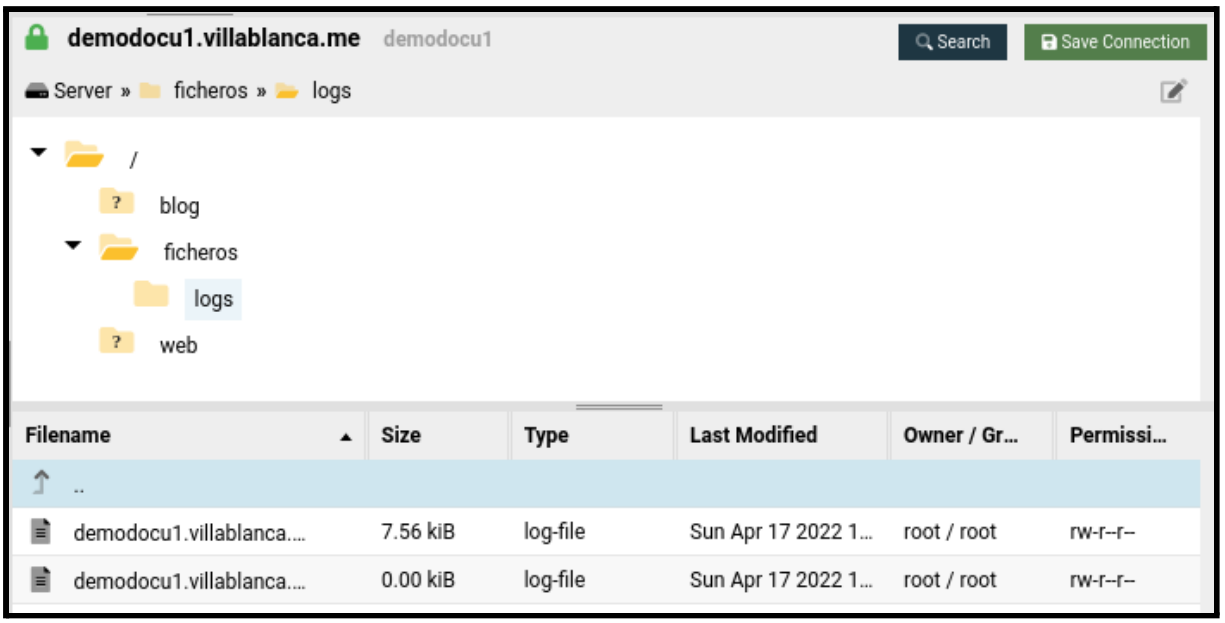
Si se selecciona que no estamos ante un sitio de pruebas, el certificado es completamente confiable:



Respecto a la conexión mediante SFTP, se indican los datos recibidos por el usuario en el mensaje de correo electrónico:

[77] https://letsencrypt.org/docs/staging-environment/

Una vez realizada la conexión, se pueden ver los directorios y los archivos. Tal y como se puede ver en la siguiente imagen, el usuario está “encerrado” en su directorio home:



5.3.3.6. - Modificar contraseña de usuario

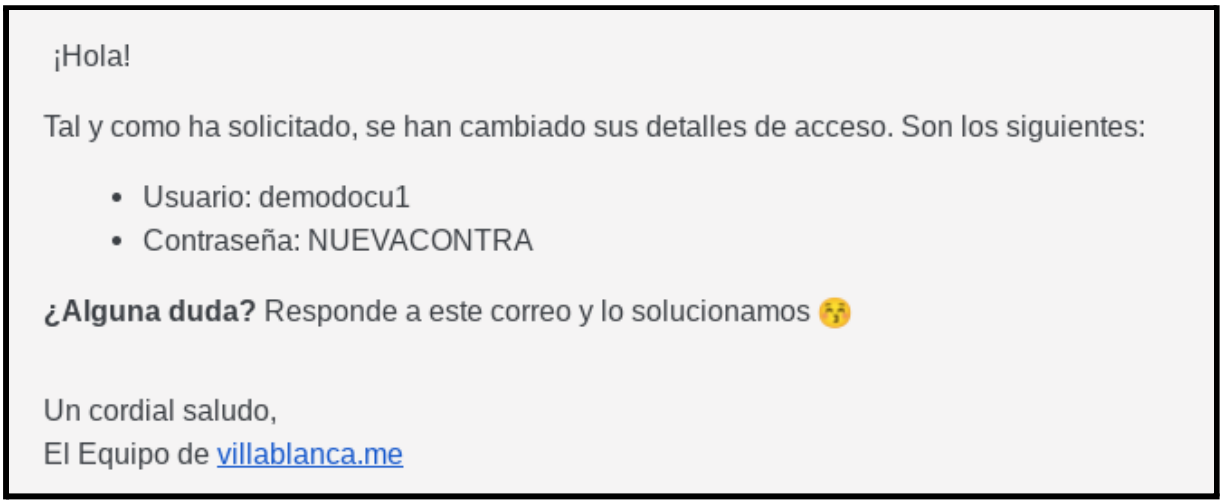
Por diversos motivos puede ser necesario un cambio en la contraseña de usuario. Nótese que estos cambios únicamente serán aplicados a la contraseña del usuario en el sistema, que es también la que se utiliza para el inicio de sesión mediante SFTP. Para el cambio de contraseñas utilizadas en las Bases de datos, así como en las aplicaciones instaladas, se seguirán otros métodos que quedan fuera de este procedimiento.

Como en anteriores ocasiones, se iniciará el script de gestión. Una vez en el menú, bastará con seleccionar la opción número 5, modificar usuarios. Una vez seleccionada la opción, nos mostrará un listado de usuarios, de entre los que debemos seleccionar uno, indicándolo en la terminal. El sistema comprueba que efectivamente, el usuario introducido existe en el servidor. Si existiera, solicita al

operador introducir doblemente la nueva contraseña, así como el correo electrónico del cliente:

```
¿Qué usuario deseas modificar? demodocu1
Introduce una nueva contraseña para el usuario demodocu1: NUEVACONTRA
Introduce de nuevo la contraseña para el usuario demodocu1: NUEVACONTRA
¡Contraseña actualizada!
Pulse cualquier tecla para continuar
Indique el correo electrónico del cliente: demodocu1@glez.tk
{"messageId":"<202204171840.25881792928@smtp-relay.mailin.fr>"}¡Listo!
Pulse cualquier tecla para volver al menú
```

El mensaje de correo electrónico que recibe el cliente es el siguiente. A su vez, se encuentra disponible una impresión del mensaje en formato PDF en este enlace[78].


¡Hola!

Tal y como ha solicitado, se han cambiado sus detalles de acceso. Son los siguientes:

- Usuario: demodocu1
- Contraseña: NUEVACONTRA

¿Alguna duda? Responde a este correo y lo solucionamos 😗

Un cordial saludo,
El Equipo de villablanca.me

5.3.3.7. Listar aplicaciones de usuario

Para listar las aplicaciones que un usuario en concreto tiene instaladas, basta con seleccionar la opción 32 en el menú. Acto seguido, se muestra al operador una lista con todos los usuarios dados de alta en el sistema. El usuario debe introducir el identificador, que el script comprobará. El listado en vertical y con una entrada por línea está hecho a propósito para facilitar copiar y pegar el usuario.

Al introducir el usuario, el script busca en un archivo especial:

```
/root/app_list/${usuario_a_listar_apps}
```

[78] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/3-webservice-evidences/correo_password_changed.pdf

Para mantener el listado de aplicaciones que tiene instaladas un usuario, se usa una especie de lista binaria. Se definirá un 1 o un 0 según la tenga instalada o no, respectivamente. Opciones:

- (Pos1) Sitio web estático
- (Pos2) WordPress
- (Pos3) PrestaShop

```
-----------------------------------------------
| PrestaShop | WordPress | Sitio web estático |
-----------------------------------------------
|     0/1    |     0/1   |        0/1         |
-----------------------------------------------
```

De esta forma un usuario que tenga el sitio web estático y una instalación de PrestaShop, tendrá el número 101 en este archivo. Si únicamente tiene instalado el sitio web estático, el 001. Si tiene instalado WordPress y el sitio web estático, el 011.

Esta función tiene varias opciones de funcionamiento:

- Modo ***silent***, no muestra el header del script, pide usuario y comprueba su existencia, retorna el valor binario de forma que puede ser recuperado en otra función. No muestra tabla de resultado.
- Modo ***bonito***, muestra el header del script, pide el usuario al operador y comprueba su existencia. Muestra una tabla con el resultado de la búsqueda pero no retorna el valor binario a la función que ha llamado a esta.
- Modo ***tabla***, no muestra el header del script, pide el usuario al operador y comprueba su existencia. Muestra la tabla como resultado y devuelve el valor binario a la función que la ha llamado.

Los diferentes modos de llamar a esta función la dotan de gran versatilidad. Se selecciona una u otra función indicando la palabra clave a través del parámetro posicional `${1}`, y el usuario a listar a través del parámetro posicional `${2}`.

Por algunos problemas con el *return* de bash, se han aplicado algunos cambios, disponibles aquí[79].

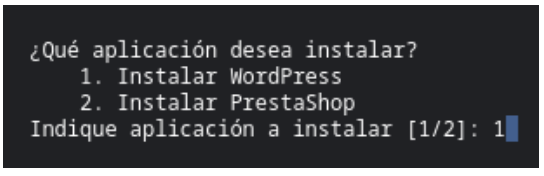
[79] https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/issues/40

A continuación se puede ver una captura de pantalla del resultado de esta función llamada desde el menú, con la opción de funcionamiento *bonito*.

5.3.3.8. Añadir aplicación: WordPress

Para añadir una aplicación a un usuario, basta con introducir el código 31 en el selector del menú. Acto seguido, listará los usuarios dados de alta en el sistema y solicitará al agente indicar el usuario sobre el que quiere trabajar.

Internamente, el proceso que se realiza es el de listar las aplicaciones. Con el resultado binario que devuelve la función *app_list*, luego el script comprueba si se está intentando instalar una aplicación ya instalada. En este caso, el propio script cancelaría la operación, notificando al agente.



En el propio script de adición de aplicaciones se genera la contraseña para el usuario. Con el usuario de trabajo y la contraseña generada se llama a dos funciones: *crear_wp* y *config_wp*.

Con la primera función llamada, *crear_wp*, es dado de alta el usuario y la base de datos, así como los permisos del primero sobre la base de datos en el motor de bases de datos. Al igual que durante la creación de los sitios estáticos, se crean los archivos de registro y se enlazan en la carpeta *ficheros/logs* del usuario. Se descarga la plantilla de configuración desde este enlace[80] y son sustituidos los valores necesarios utilizando el comando *sed*.

```
<VirtualHost *:80>
    ServerAdmin USER-TO-CHANGE@localhost
    ServerName blog.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN
    DocumentRoot /var/www/USER-TO-CHANGE/blog
    ErrorLog /var/log/apache2/blog.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN.log
    CustomLog /var/log/apache2/blog.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-access.log combined
    AssignUserID USER-TO-CHANGE USER-TO-CHANGE
    <Directory /var/www/USER-TO-CHANGE/blog/>
        AllowOverride All
    </Directory>
</VirtualHost>
```

```
wget -qO /etc/apache2/sites-available/wp_${1}.conf
https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/templates%20and%20misc/wp_virtualhost

sed -i "s/USER-TO-CHANGE/${1}/g" "/etc/apache2/sites-available/wp_${1}.conf"

sed -i "s/GLOBAL-BASE-DOMAIN/${global_base_domain}/g"
"/etc/apache2/sites-available/wp_${1}.conf"
```

Se activa el sitio virtual y se reinicia el servidor web Apache. En una versión inicial, con cada instalación de WordPress se descargaba el archivo desde la página web oficial. Esto era poco óptimo y costoso. Por tanto, se ha trabajado en que únicamente descargue y extraiga el archivo comprimido en caso de que no exista previamente el la ruta temporal.

Copia el contenido extraído en la ubicación de destino, dentro del directorio del usuario y aplica los permisos y configuración de propiedad correspondientes.

La segunda función, *config_wp*, se encarga de, una vez está creado el sitio virtual en el servidor web y los archivos de WordPress, añadir la configuración de la base

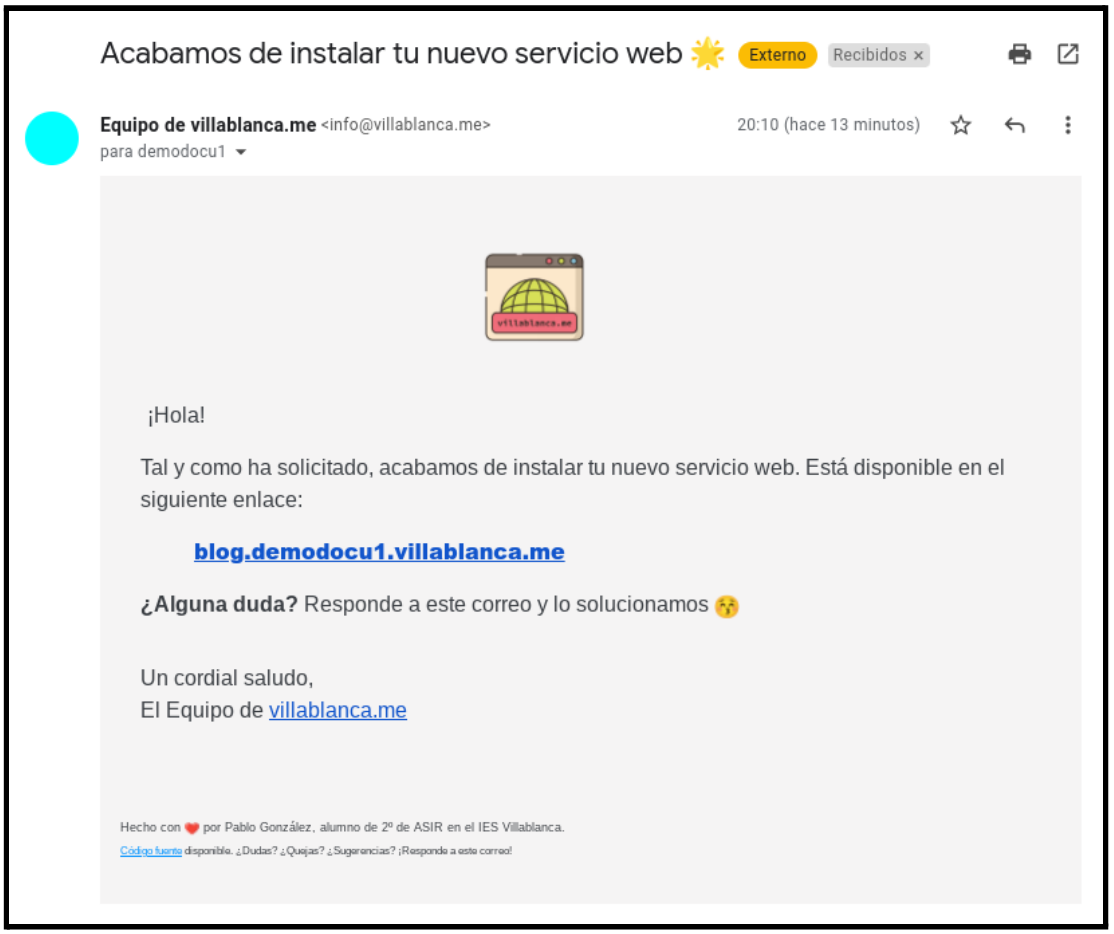
[80] https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/templates%20and%20misc/wp_virtualhost

de datos al archivo *wp-config.php*. También se usa la API de WordPress[81] para añadir los *salts* para asegurar la instalación. Los valores son también sustituidos utilizando el comando sed.

A continuación, se añade el registro DNS y se genera el certificado TLS.

```
Comprobando la resulución del dominio blog.demodocu1.villablanca.me
 Por favor, espera...
¡Excelente! El registro creado es válido

Pulse cualquier tecla para continuar el proceso.
```

Es solicitado un correo electrónico y a esta dirección se envía una notificación de la instalación y los detalles de acceso.

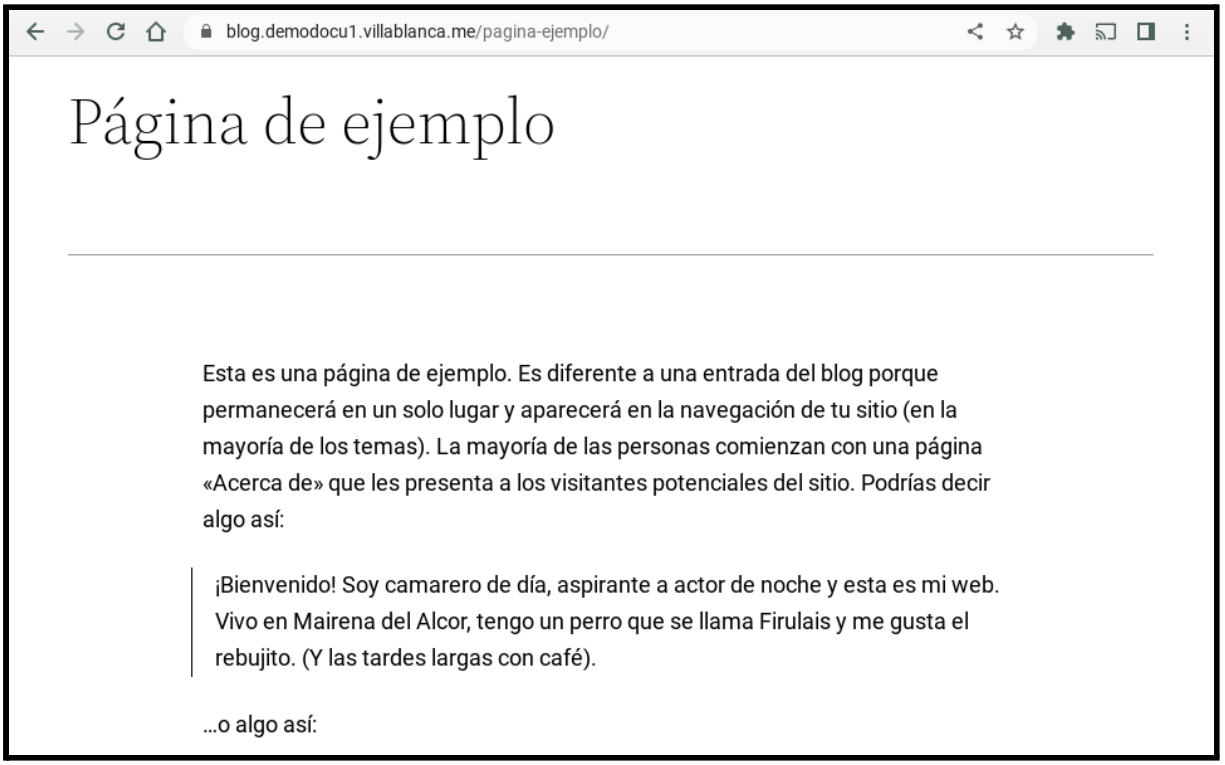


A su vez, se puede acceder a la impresión en PDF del correo recibido desde este enlace[82].

[81] https://api.wordpress.org/secret-key/1.1/salt/
[82] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/3-webservice-evidences/correo_wp_installed.pdf

Cuando el usuario entra al enlace que ha recibido por correo electrónico, ya puede ver la interfaz de configuración:

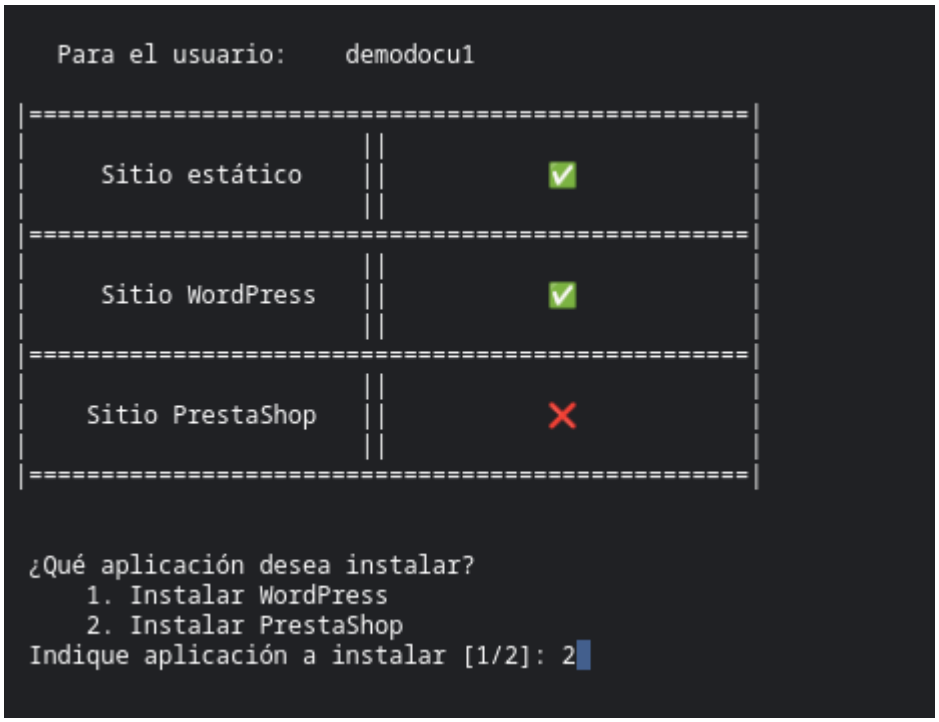
La instalación de WordPress es funcional, a continuación se muestra la página de ejemplo:



5.3.3.9. Añadir aplicación: Prestashop

El proceso de instalación del servicio de Prestashop es similar al que se realiza para la instalación de WordPress.

Puesto que es una función creada recientemente, es más avanzada y todas las operaciones necesarias se realizan con una única llamada a la función *install_prestashop*, facilitando los datos del usuario para el cual hay que instalar la aplicación.



Como se puede ver en la imagen, puesto que es una instalación posterior a la de WordPress, el programa ya reconoce la instalación de WordPress e informa al agente.

Se crea el directorio *tienda* dentro del *home* del usuario y se aplican los permisos correspondientes. Acto seguido, al igual que con el resto de aplicaciones, se crean los archivos de registro y se enlazan. Se descarga desde este enlace[83] el archivo de configuración del sitio virtual, que también se puede ver a continuación en este documento, se sustituyen los valores clave, se activa el sitio y se reinicia el servidor web.

[83] https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/templates%20and%20misc/tienda_virtualhost.txt

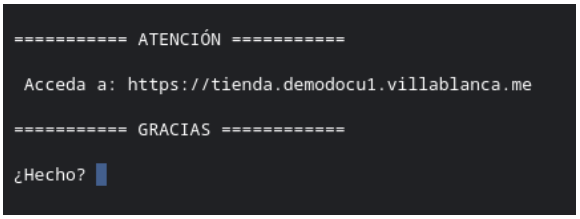
```
<VirtualHost *:80>
    ServerAdmin admin@localhost
    ServerName tienda.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN
    DocumentRoot /var/www/USER-TO-CHANGE/tienda

    <Directory /var/www/USER-TO-CHANGE/tienda>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog /var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-tienda.log
    CustomLog
/var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-tienda-access.log combined
    AssignUserID USER-TO-CHANGE USER-TO-CHANGE

</VirtualHost>
```

Se crean la base de datos y el usuario, configurando también los permisos de este. Se comprueba si ya existen los archivos de instalación de PrestaShop. De no existir, son descargados y descomprimidos en una ruta temporal.

Los archivos son copiados desde la citada ruta temporal hasta su destino final, en el directorio *tienda* del usuario sobre el que se está trabajando. Se aplican los permisos y propiedades correspondientes a los usuarios para asegurarse de que esta nueva instalación no afectará negativamente al acceso mediante SFTP, por los requisitos del proceso de *chroot*.

Se añade la entrada DNS y se genera el certificado para el nuevo sitio web. Como diferencia respecto a WordPress, no es posible indicar la conexión con la base de datos en ningún archivo, o al menos se desconoce esta posibilidad. Para solventarlo, el propio script guía al agente para completar la instalación de forma satisfactoria, indicando qué valores ha de indicar en cada pantalla de configuración.

Veamos cómo guía el script al agente durante la instalación:

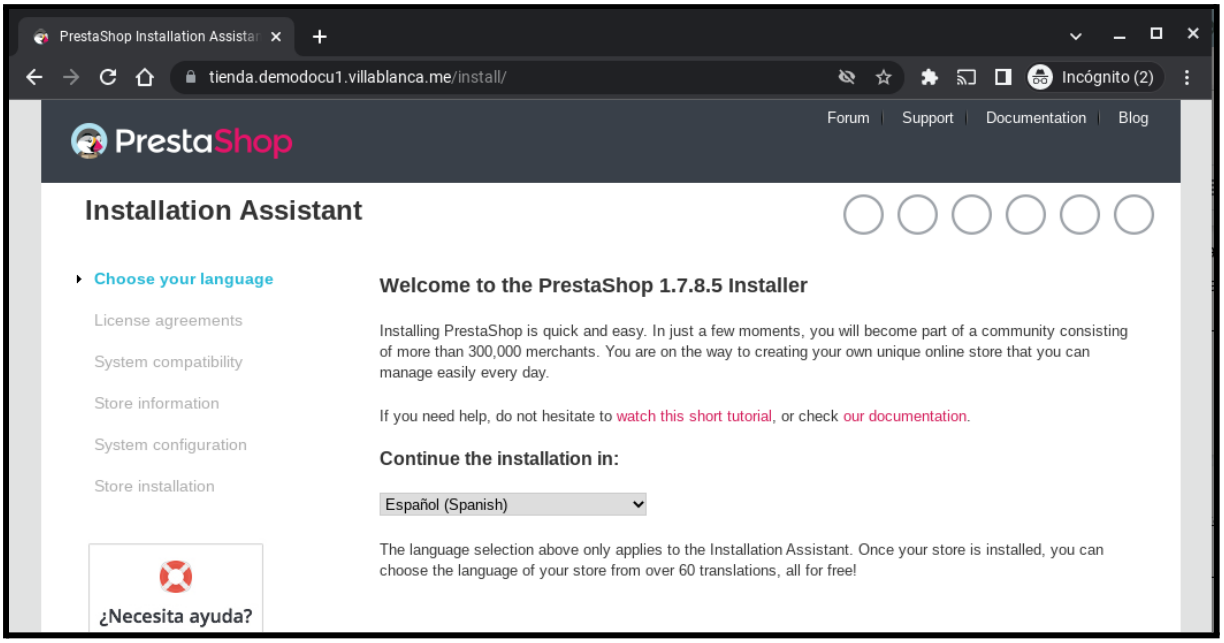


Como vemos en la imagen anterior, envía al agente a la página web en la que se acaba de instalar Prestashop. Solicita confirmación para continuar.

```
========== ATENCIÓN ==========
 Idioma: Español (Spanish)
========== GRACIAS ===========
¿Hecho?
```

Indica al agente que seleccione *Español (Spanish)* para el proceso de instalación

En la siguiente imagen podemos ver el proceso de instalación en este momento:




```
========== ATENCIÓN ==========
 Acepta términos y pulsa 'Siguiente'
========== GRACIAS ===========
¿Hecho?
```

Informa al agente de la necesidad de aceptar los términos de Prestashop.

Se entiende que el usuario final ha sido informado de los términos de este y los ha aceptado de forma previa a la instalación.

A continuación, se muestran al agente los datos más importantes de todo el proceso de instalación de Prestashop.

Entre estos datos están los relativos a la base de datos y las credenciales que usará el usuario para iniciar sesión en la plataforma de administración web. A su vez, se muestran datos como el nombre de la tienda (que el propio usuario podrá modificar

más adelante desde su panel), si se desea utilizar SSL para las conexiones, el sector de actividad y si añadir datos de demostración.

```
========== ATENCIÓN ==========

 Nombre:
Tienda de demodocu1

 Actividad: Otra

 Datos demostración: 'Sí'

 Activar SSL: Sí

 Correo del usuario:
demodocu1@glez.tk

 Contraseña de usuario:
+n3s63pZRbkABh28

========== GRACIAS ===========

¿Hecho?
```

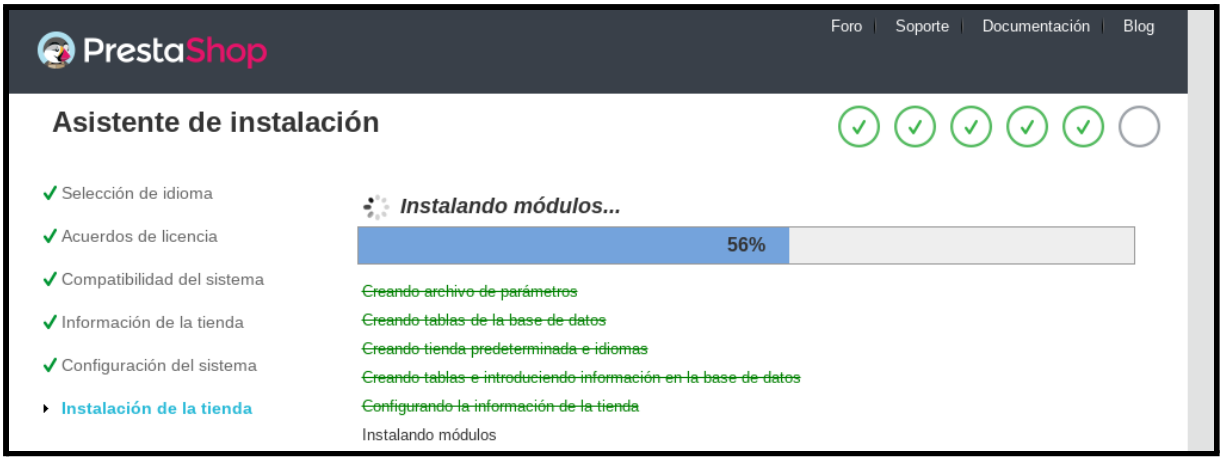
Los datos que el agente ha de copiar desde la terminal y pegar en la web no están indentados ni separados por espacios del margen para facilitar el copiado con una simple selección de texto inteligente desde el programa cliente de SSH.

Se muestran en este documento las credenciales pues tienen carácter demostrativo. La conexión a la base de datos únicamente puede realizarse desde el propio equipo (*localhost*).

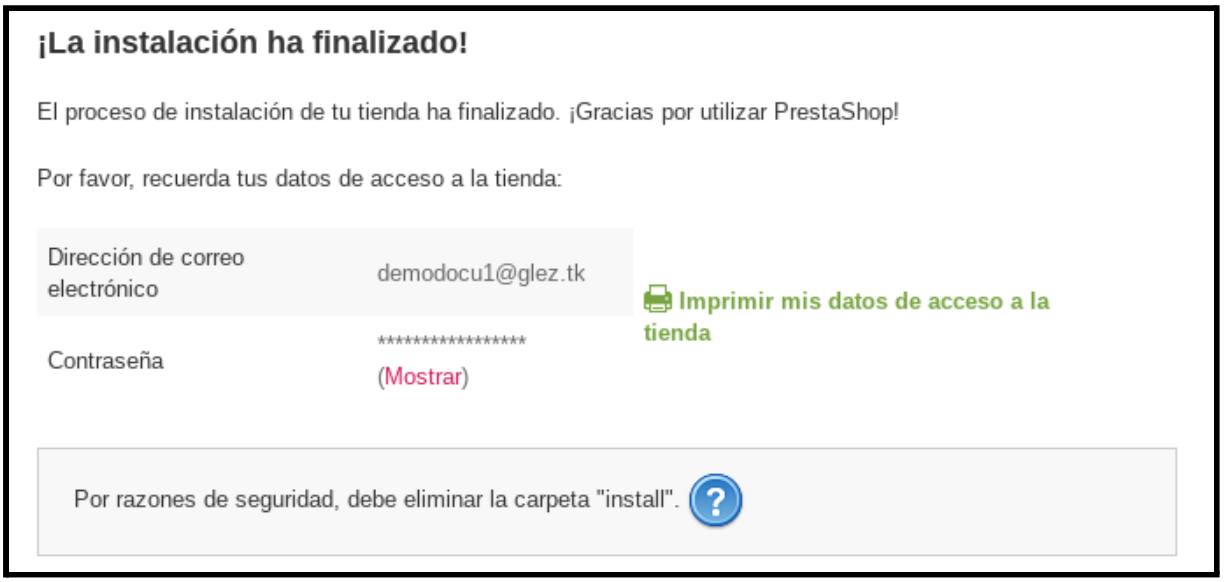
Al introducir los detalles de la conexión con la base de datos, la propia interfaz web da la posibilidad al agente de comprobar la conexión. Esto es realmente útil para que el agente se cerciore que ha copiado y pegado correctamente los valores mostrados en la terminal, sin ninguna clase de error. En cualquier caso, si los valores no fueran aceptados por la interfaz, se podría acceder al motor de base de datos para ver el motivo.

```
========== ATENCIÓN =========

 BDD:
tienda_demodocu1

 Usuario:
demodocu1_tienda

 Contraseña:
8v9kZR4eDFDeLl1P

========== GRACIAS ==========

¿Hecho?
```

Se continuará automáticamente con la instalación de forma automática, pudiendo tardar este algunos minutos.



La pantalla de confirmación es la siguiente:

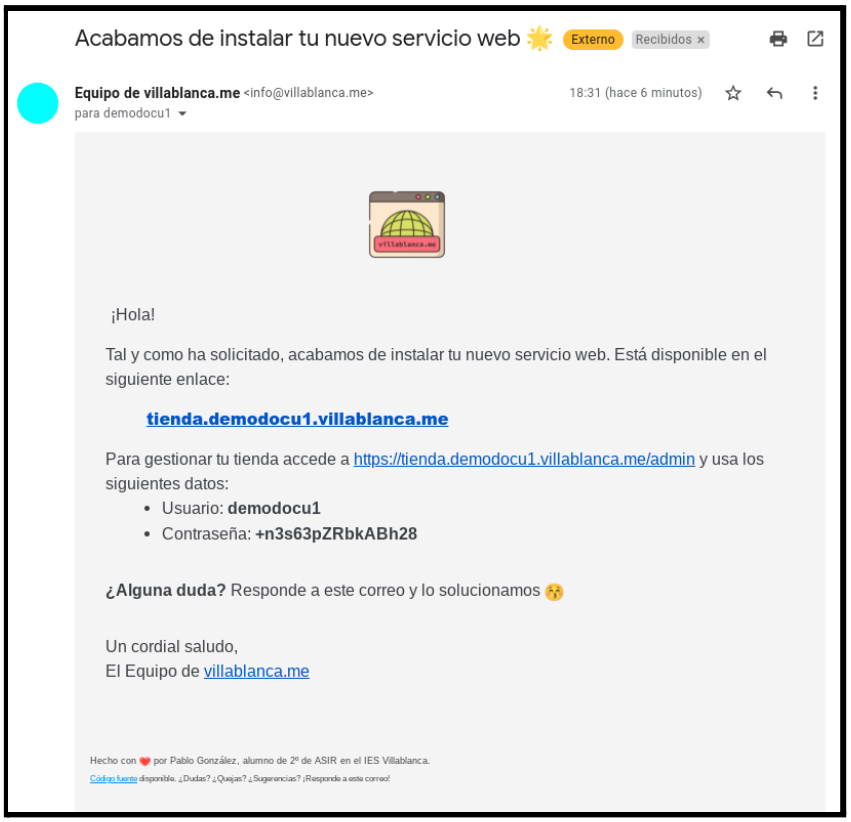


Como se puede leer en esta, la carpeta “install” debe ser eliminada. Esta carpeta se encuentra en el path de instalación de PrestaShop. Se puede obtener más información sobre esta necesidad en este enlace[84]. El script desarrollado ya lo hace después de la última confirmación del agente:

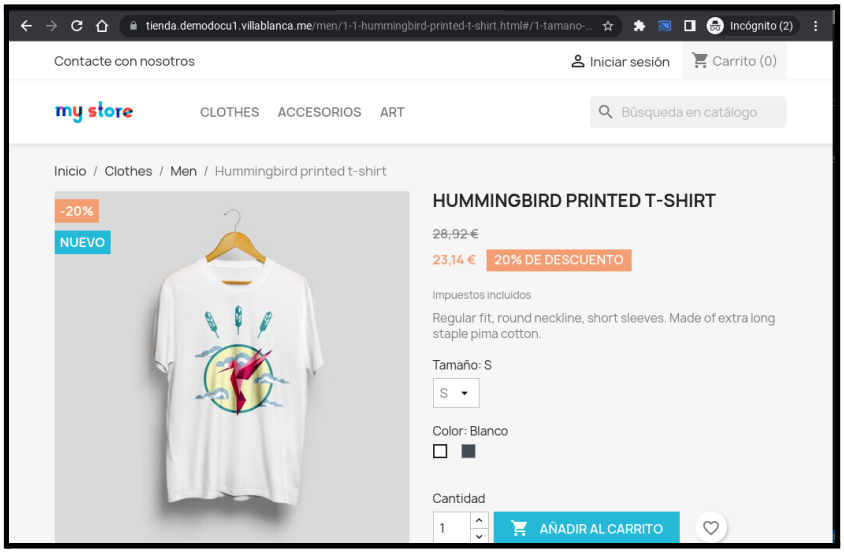
```
read -p "Pulsar al término de la instalación " trash

rm -rf /var/www/${usuario}/tienda/install/
```

[84] https://doc.prestashop.com/display/PS17/Instalar+PrestaShop#InstalarPrestaShop-C%C3%B3mocompletarlainstalaci%C3%B3n

Como con todas las instalaciones y cambios, un mensaje de correo electrónico es enviado al usuario con la confirmación de la instalación y sus datos de acceso. El correo electrónico recibido es similar al siguiente y la impresión en PDF puede ser descargada desde este enlace[85]:



La tienda es funcional, pudiendo ser accedidos los artículos de demostración:



[85] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/3-webservice-evidences/correo_ps_installed.pdf

5.3.3.10. Borrar usuarios

Para el borrado de los usuarios del servidor, existen dos posibilidades: un borrado *suave*, que programa la eliminación del usuario y sus datos; y un borrado *duro*, que elimina definitivamente el usuario y sus datos.

El primero, el borrado *suave*, es el que se debe usar en los casos de producción. El segundo modo de borrado, el borrado *duro*, se ha creado más como necesidad interna, pues durante el proceso de desarrollo ha sido necesaria la eliminación de muchos usuarios de pruebas.

Para la programación *suave:*

- Se deshabilita el usuario, para prevenir el inicio de sesión:

```
usermod -L ${usuario_a_borrar}
```

- Se programa la eliminación del usuario en 30 días:

```
at now + 30 days "userdel -f ${usuario_a_borrar}"
```

- Se deshabilitan los sitios y el acceso del usuario a las bases de datos:

```
a2dissite ${usuario_a_borrar}.conf > /dev/null
a2dissite ${usuario_a_borrar}-le-ssl.conf > /dev/null
a2dissite wp_${usuario_a_borrar}.conf > /dev/null
a2dissite wp_${usuario_a_borrar}-le-ssl.conf > /dev/null
a2dissite tienda_${usuario_a_borrar}.conf > /dev/null
a2dissite tienda_${usuario_a_borrar}-le-ssl.conf > /dev/null
mysql -e "REVOKE ALL PRIVILEGES ON wp_${usuario_a_borrar}.* FROM
          ${usuario_a_borrar};"
mysql -e "REVOKE ALL PRIVILEGES ON ${usuario_a_borrar}_tienda.* FROM
          ${usuario_a_borrar}_tienda'@'localhost';"
systemctl reload apache2
```

- Se programa la eliminación de las bases de datos y el usuario de estas:

```
echo "rm -Rf /var/www/${usuario_a_borrar}" | at now + 30 days
echo "mysql -e 'DROP DATABASE IF EXISTS wp_${usuario_a_borrar};'" | at now + 30
      days
echo "mysql -e 'DROP DATABASE IF EXISTS ${usuario_a_borrar}_tienda;'" | at now +
      30 days
```

```
echo "mysql -e 'DROP USER IF EXISTS ${usuario_a_borrar};'" | at now + 30 days
echo "mysql -e 'DROP USER IF EXISTS ${usuario_a_borrar}_tienda;'" | at now + 30
      days
```

- Por último, se informa al agente de la eliminación exitosa y le retorna al menú de la utilidad:

```
echo "${usuario_a_borrar}, sus sitios y accesos hasn sido deshabilitados
      correctamente"

echo "${usuario_a_borrar} y sus sitios han sido programados para eliminación en
      30 días."

read -p "Pulse intro para volver al menú" caca

menu
```

Para la variación de borrado *fuerte*, el proceso es similar, pero sin posponer la eliminación del usuario y los datos 30 días. Directamente son eliminados.

La interfaz es realmente sencilla. Una vez seleccionada una de las dos opciones, la utilidad lista los usuarios dados de alta en el sistema.

Acto seguido, solicita al usuario introducir el identificador del usuario que desea eliminar, y procede con su eliminación.

5.3.4. - Script en Python para limpieza registros en Cloudflare

El script también puede ser descargado desde este enlace[86], a su vez se muestra en este documento.

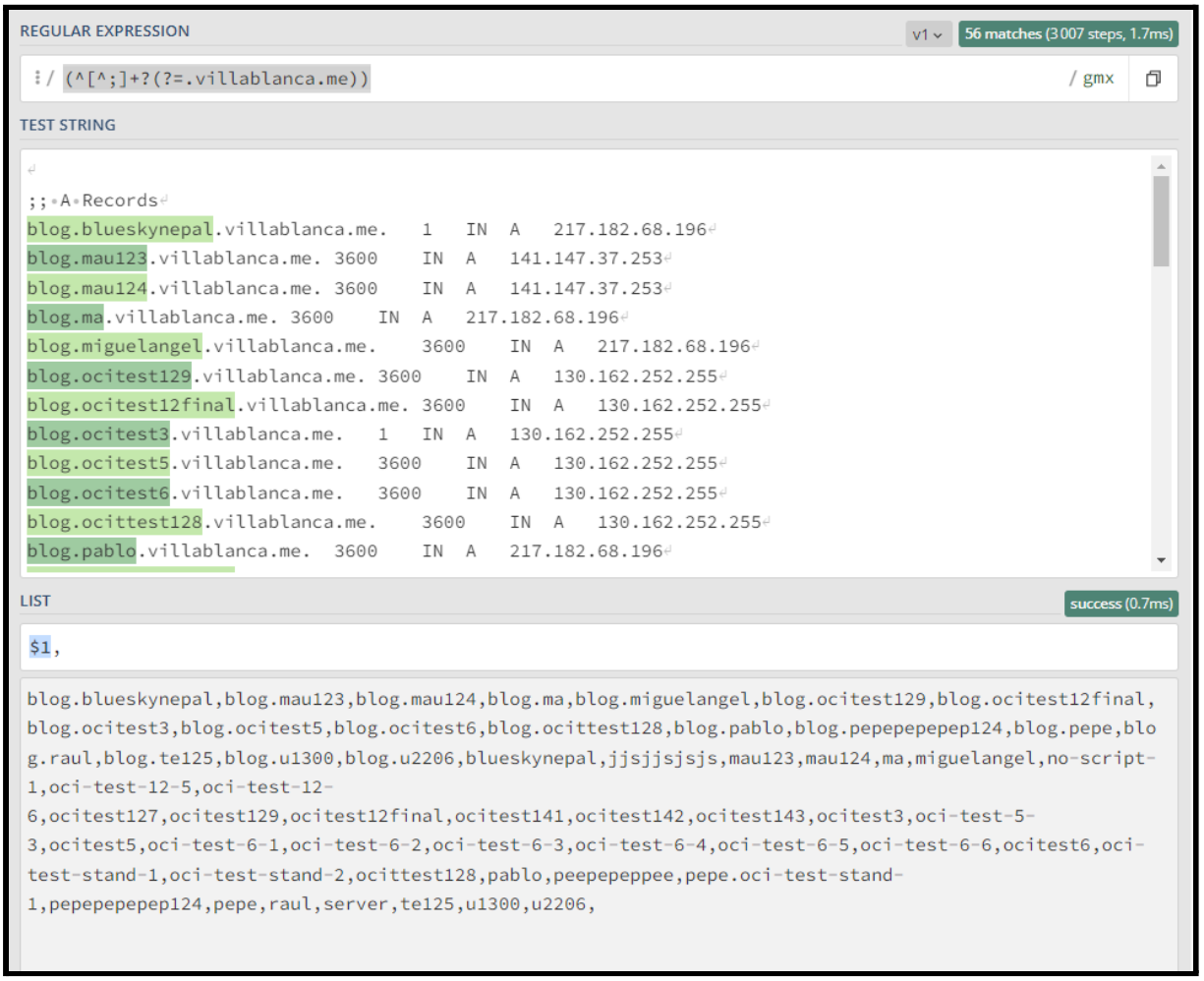
Para utilizarlo, basta con descargar desde Cloudflare el archivo de zona. Este es un archivo en formato BIND con todos los registros.

Una vez hecho esto, aplicaremos la siguiente expresión regular para obviar el resto de datos:

```
(^[^;]+?(?=.villablanca.me))
```

[86] https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/blob/main/templates%20and%20misc/cloudflare-cleaner.py

En esta página web[87] se puede ver el resultado real, sobre datos reales. Se adjunta también la siguiente captura como referencia:



En la parte inferior de la pantalla ya puede ser vista la lista de los registros. Una vez obtenidos los registros, basta con ejecutar en bash:

```
for sub in {blog.blueskynepal,blog.mau123,blog.mau124,blog.ma,blog.miguelangel}
 do
    python3 cloudflare-cleaner.py  villablanca.me $sub
 done
```

La parte destacada en verde sobre el fragmento de código anterior es lo que habría que sustituir con el resultado obtenido de la aplicación de la expresión regular.

[87] https://regex101.com/r/cqc8al/1

En tanto al script, es muy sencillo, como puede verse a continuación. Usa el *wrapper* de la API[88] de Cloudflare para Python

```
#!/bin/env python3

import CloudFlare
import os
import sys

def main():
    try:
        zone_name = sys.argv[1]
        dns_name = sys.argv[2]
    except IndexError:
        exit('usage: example_delete_zone_entry.py zone dns_record')

    cf = CloudFlare.CloudFlare(token='TOKEN') # AQUÍ TOKEN
    try:
        params = {'name':zone_name}
        zones = cf.zones.get(params=params)
    except CloudFlare.exceptions.CloudFlareAPIError as e:
        exit('/zones %d %s - api call failed' % (e, e))
    except Exception as e:
        exit('/zones.get - %s - api call failed' % (e))

    if len(zones) == 0:
        exit('/zones.get - %s - zone not found' % (zone_name))

    if len(zones) != 1:
        exit('/zones.get - %s - api call returned %d items' % (zone_name, len(zones)))

    zone = zones[0]
    zone_id = zone['id']
    zone_name = zone['name']

    print('ZONE:', zone_id, zone_name)
    try:
        params = {'name':dns_name + '.' + zone_name}
        dns_records = cf.zones.dns_records.get(zone_id, params=params)
    except CloudFlare.exceptions.CloudFlareAPIError as e:
        exit('/zones/dns_records %s - %d %s - api call failed' % (dns_name, e, e))

    found = False
    for dns_record in dns_records:
        dns_record_id = dns_record['id']
        dns_record_name = dns_record['name']
        dns_record_type = dns_record['type']
        dns_record_value = dns_record['content']
        print('DNS RECORD:', dns_record_id, dns_record_name, dns_record_type, dns_record_value)
        try:
            dns_record = cf.zones.dns_records.delete(zone_id, dns_record_id)
            print('DELETED')
        except CloudFlare.exceptions.CloudFlareAPIError as e:
            exit('/zones.dns_records.delete %s - %d %s - api call failed' % (dns_name, e, e))
        found = True

    if not found:
        print('RECORD NOT FOUND')

    exit(0)

if __name__ == '__main__':
    main()
```

[88] https://github.com/cloudflare/python-cloudflare

5.4. Servicio VoIP 📞

Para el servicio VoIP se ha elegido el proveedor de servicios IaaS clouding.io[89]. Iniciar un servidor es realmente sencillo y rápido. En la sección 5.2.1.3. Definición del servidor virtual se encuentra toda la información paso a paso.

Para esta sección del proyecto se ha alojado el servidor en clouding.io, que ha colaborado con el proyecto patrocinando los servidores utilizados.

La solución que se instalará es FreePBX. Esta solución[90] *Open Source* permite mejorar el ya excelente sistema de gestión SIP Asterisk[91]. De hecho, FreePBX utiliza Asterisk como backend, simplemente añadiendo una interfaz web de gestión web y ciertas ventajas como módulos y gestión de directorios simplificadas. Detrás de ambas soluciones se encuentra la empresa Sangoma[92]

5.4.1. Instalación de utilidades

5.4.1.1 Instalación de Asterisk

Antes de realizar cualquier acción, refrescaremos los paquetes disponibles en los repositorios con el siguiente comando:

```
apt update -y
```

Los paquetes que en este momento debemos instalar son varios. Es probable que algunos ya estén disponibles en nuestro sistema, pero los incluiremos para

[89] https://clouding.io/nosotros
[90] https://www.freepbx.org/
[91] https://www.asterisk.org/
[92] https://www.sangoma.com/

asegurarnos de que tenemos instalada la última versión. Otros, como los necesarios para la compilación, pueden ser eliminados posteriormente.

```
apt-get install unzip git gnupg2 curl libnewt-dev libssl-dev libncurses5-dev
subversion libsqlite3-dev build-essential libjansson-dev libxml2-dev uuid-dev
                                subversion -y
```

Al igual que ocurre con algunas distribuciones de sistemas operativos[93], Asterisk se presenta en versiones “LTS”, *Long Term Support* y *Standard*. En el momento de realizar la instalación la última versión es la 19, siendo la versión 18 la última LTS.

En un entorno real, sería más conveniente usar las versiones LTS, pero en este caso instalaremos la 19 para disfrutar las últimas funcionalidades. Descargamos en este momento el código, también lo extraemos:

```
                                    wget
https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-19-current.tar.gz

                          tar zxf asterisk-19-current.tar.gz
```

Si en este momento accedemos a la carpeta, veremos una serie de directorios:

```
                                cd asterisk-19.3.2/
                                        ls
```

addons	codecs	images	pbx
agi	config.guess	include	phoneprov
apps	config.log	install-sh	README-addons.txt
asterisk-19.3.2-summary.html	configs	LICENSE	README.md
asterisk-19.3.2-summary.txt	config.status	main	README-SERIOUSLY.bestpractices.md
autoconf	config.sub	Makefile	res
bootstrap.sh	configure	Makefile.moddir_rules	rest-api
bridges	configure.ac	Makefile.rules	rest-api-templates
BSDmakefile	contrib	makeopts	sample.call
BUGS	COPYING	makeopts.in	sounds
build_tools	CREDITS	menuselect	static-http
cdr	default.exports	menuselect.makedeps	tests
cel	defaults.h	menuselect.makeopts	third-party
ChangeLog	doc	menuselect-tree	UPGRADE.txt
CHANGES	formats	missing	utils
channels	funcs	mkinstalldirs	Zaptel-to-DAHDI.txt

Ejecutaremos en este momento 2 scripts incluidos en el código, que nos ayudarán a instalar los códecs MP3 y los prerrequisitos. Respectivamente son:

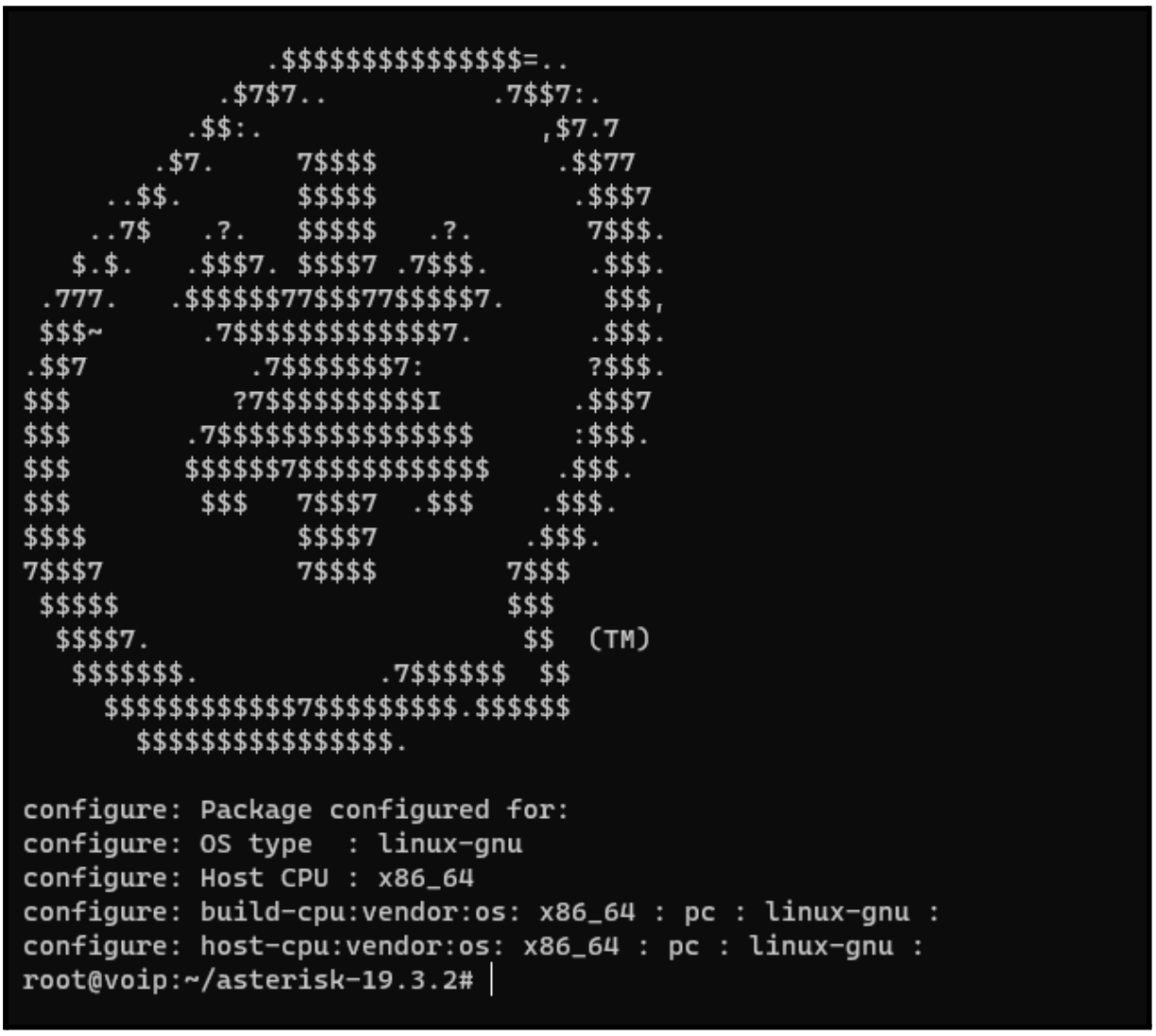
[93] https://ubuntu.com/blog/what-is-an-ubuntu-lts-release

```
contrib/scripts/get_mp3_source.sh

contrib/scripts/install_prereq install
```

Ejecutaremos el siguiente comando, que configurará las fuentes y la configuración para compilar para nuestro sistema:

```
./configure
```

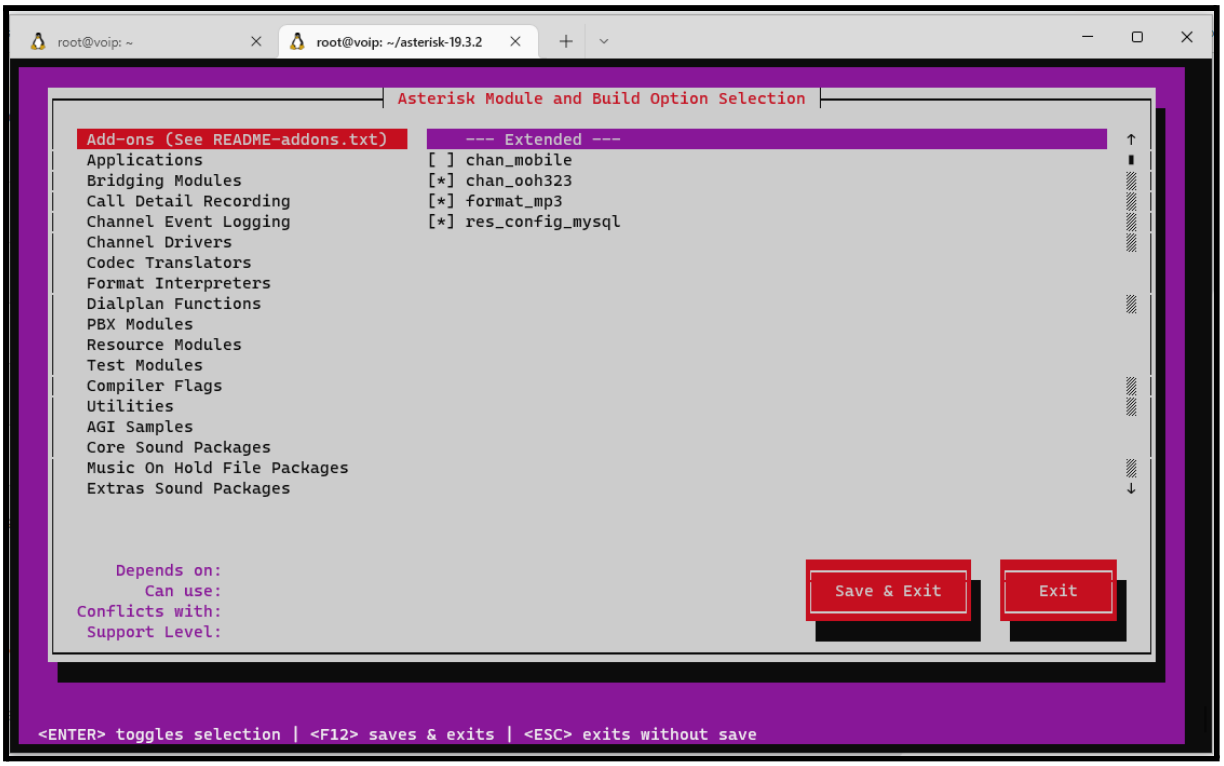
La instalación terminará con una pantalla similar a la siguiente:



Configuraremos en este momento los módulos propios de Asterisk (estos no son los módulos de FreePBX) con su utilidad. Para ello ejecutamos:

```
make menuselect
```

Aparecerá una ventana similar a la siguiente, donde usaremos las flechas para desplazarnos por los menús y la tecla “Intro” para seleccionar/deseleccionar las diferentes opciones:



Para la instalación de nuestro sistema, se han seleccionado los siguientes módulos. En una primera instalación no estaban todos los incluidos en esta lista. Se basa en diversos tutoriales[94] [95] [96] [97] [98] [99] [100] y aprendizaje prueba-error:

chan_ooh323 CORE-SOUNDS-ES-ES*	format_mp3 app_macro	MOH-OPSOUND-*

Pueden ser necesarios otros paquetes/módulos dependiendo de nuestras necesidades. Siempre es posible volver a ejecutar el menú de configuración y

[94] https://www.atlantic.net/vps-hosting/how-to-install-asterisk-and-freepbx-on-ubuntu-20-04/
[95] https://websiteforstudents.com/how-to-install-freepbx-on-ubuntu-18-04-16-04/
[96] https://wiki.freepbx.org/display/FOP/Installing+FreePBX+14+on+Ubuntu+18.04
[97] https://wiki.freepbx.org/display/FPG/Configuring+Your+PBX
[98] https://www.linuxhelp.com/how-to-install-asterisk-16-lts-on-ubuntu-21-04
[99] https://gist.github.com/kolosek/f0d1952f784f7f164db145497ce155b6
[100] https://wiki.asterisk.org/wiki/display/AST/Using+Menuselect+to+Select+Asterisk+Options

recompilar el ejecutable de asterisk para que los cambios realizados en el *menuselect* se guarden.

Una vez hayamos completado los cambios, los guardamos pulsando F12. Veremos el siguiente mensaje en nuestra terminal:

```
           +--------- Asterisk Build Complete ---------+
           + Asterisk has successfully been built, and +
           + can be installed by running:              +
           +                                           +
           +                make install               +
           +-------------------------------------------+
```

Sin embargo, antes de instalar asterisk debemos “buildear” el archivo ejecutable. Para ello, debemos ejecutar el siguiente comando en nuestra terminal:

```
                          make -j2
```

Una vez compilado, instalamos. También puede ser interesante instalar las configuraciones y ejemplos para facilitar la instalación:

```
                        make install

                        # Opcionales
                        make samples
                         make config
                           ldconfig
```

5.4.1.2. Configuración de Asterisk

Creamos el grupo y el usuario con el cual se ejecutará Asterisk:

```
                      groupadd asterisk
       useradd -r -d /var/lib/asterisk -g asterisk asterisk
```

Esto es importante para evitar que el programa se ejecute con permisos de superusuario, que podrían exponer nuestro sistema.

Añadimos al grupo asterisk recién creado otros usuarios necesarios:

```
usermod -aG audio,dialout asterisk
```

Configuramos de forma correcta los permisos de los diferentes directorios:

```
chown -R asterisk.asterisk /etc/asterisk
chown -R asterisk.asterisk /var/{lib,log,spool}/asterisk
chown -R asterisk.asterisk /usr/lib/asterisk
```

Editamos el archivo nano `/etc/default/asterisk`, para indicar aquí qué usuario queremos que ejecute la aplicación:

```
/etc/default/asterisk
```

```
AST_USER="asterisk"
AST_GROUP="asterisk"
```

Para esta misma función también es necesario editar un segundo archivo:

```
/etc/asterisk/asterisk.conf
```

```
runuser = asterisk ; The user to run as.
rungroup = asterisk ; The group to run as.
```

Guardamos los cambios y relanzamos el servicio. También es interesante activarlo (“enable”) para que se ejecute de forma automática al reiniciar el servidor:

```
systemctl restart asterisk
systemctl enable asterisk
```

Podemos comprobar el estado con el siguiente comando:

```
systemctl status asterisk
```

```
● asterisk.service - LSB: Asterisk PBX
     Loaded: loaded (/etc/init.d/asterisk; generated)
     Active: active (running) since Mon 2022-05-16 19:17:15 CEST; 49min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 1041 ExecStart=/etc/init.d/asterisk start (code=exited, status=0/SUCCESS)
      Tasks: 76 (limit: 4612)
     Memory: 119.8M
     CGroup: /system.slice/asterisk.service
             └─1086 /usr/sbin/asterisk -U asterisk -G asterisk

May 16 19:17:15 voip.glez.cloud systemd[1]: Starting LSB: Asterisk PBX...
May 16 19:17:15 voip.glez.cloud asterisk[1041]:  * Starting Asterisk PBX: asterisk
May 16 19:17:15 voip.glez.cloud asterisk[1041]:    ...done.
```

Error en *radiusclient*

En el caso de que localicemos el siguiente error en los registros debemos ejecutar una serie de comandos. Esta es la entrada de log:

```
radcli: rc_read_config: rc_read_config: can't open
/etc/radiusclient-ng/radiusclient.conf: No such
file or directory
```

Para solucionarlo debemos ejecutar una serie de “seds”, que nos ayudarán a editar los archivos afectados:

```
sed -i 's";\[radius\]"\[radius\]"g' /etc/asterisk/cdr.conf

sed -i 's";radiuscfg =>
/usr/local/etc/radiusclient-ng/radiusclient.conf"radiuscfg =>
/etc/radcli/radiusclient.conf"g' /etc/asterisk/cdr.conf

sed -i 's";radiuscfg =>
/usr/local/etc/radiusclient-ng/radiusclient.conf"radiuscfg =>
/etc/radcli/radiusclient.conf"g' /etc/asterisk/cel.conf
```

5.4.1.3. Instalación de FreePBX

La interfaz web de FreePBX requiere un servidor web (instalaremos apache2), una base de datos (instalaremos MariaDB) y la versión 7.2 de PHP. Este dato es importante, pues no todos los repositorios tienen esta versión. La más común es la 7.4. Antes de nada ejecutaremos:

```
apt-get install software-properties-common –y
    add-apt-repository ppa:ondrej/php -y
                apt update
```

Instalamos los paquetes necesarios:

```
apt-get install apache2 mariadb-server libapache2-mod-php7.2 php7.2
   php-pear php7.2-cgi php7.2-common php7.2-curl php7.2-mbstring
     php7.2-gd php7.2-mysql php7.2-bcmath php7.2-zip php7.2-xml
            php7.2-imap php7.2-json php7.2-snmp
```

Una vez instalados los paquetes necesarios, descargamos y extraemos el código de FreePBX ejecutando los siguientes comandos:

```
wget http://mirror.freepbx.org/modules/packages/freepbx/freepbx-15.0-latest.tgz

tar -xvzf freepbx-15.0-latest.tgz
```

Para la gestión de paquetes se usa node.js, debemos instalarlo:

```
apt-get install nodejs -y
```

Entremos en la carpeta y ejecutamos el comando para instalar los requisitos:

```
cd freepbx

./install -n
```

```
# En una instalación satisfactoria, obtenemos la siguiente salida

   Setting specific permissions...
30690 [============================]
    Finished setting permissions
  Generating default configurations...
Finished generating default configurations
 You have successfully installed FreePBX
```

5.4.1.4. Configuración de FreePBX

Para que el servidor web también se ejecute con el usuario *asterisk* (que hemos creado anteriormente) debemos cambiar los ajustes del sitio de Apache:

```
sed -i 's/^\(User\|Group\).*/\1 asterisk/' /etc/apache2/apache2.conf

sed -i 's/AllowOverride None/AllowOverride All/'
    /etc/apache2/apache2.conf
```

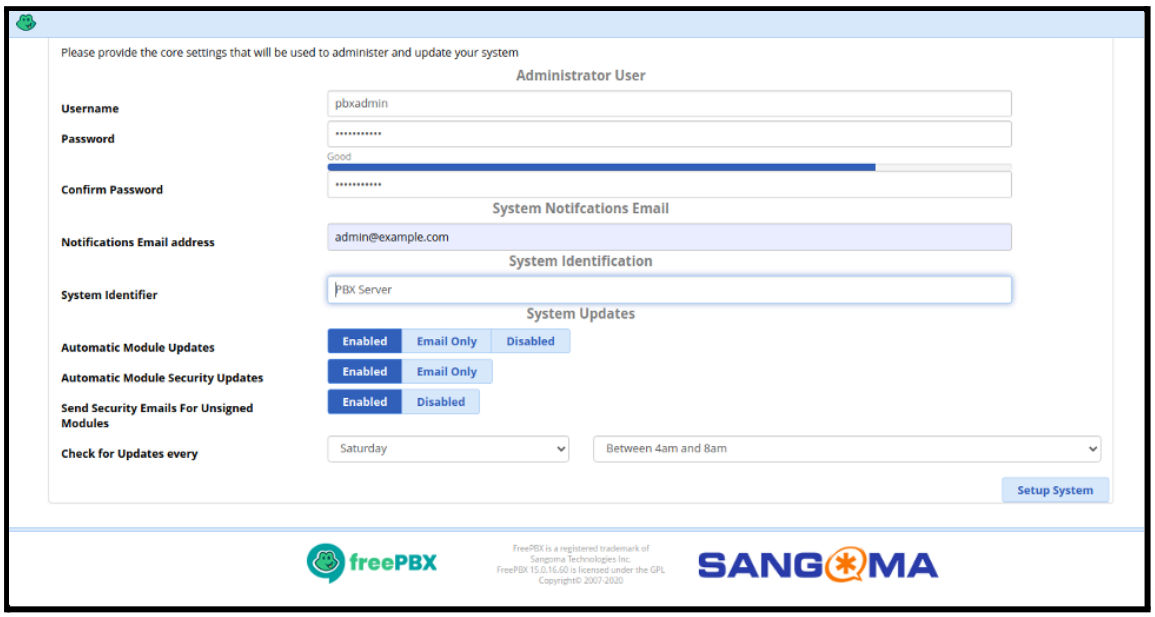
También aumentamos el tamaño máximo de subida de archivo en PHP

```
sed -i 's/\(^upload_max_filesize = \).*/\120M/'
    /etc/php/7.2/apache2/php.ini
sed -i 's/\(^upload_max_filesize = \).*/\120M/'
    /etc/php/7.2/cli/php.ini
```

Activamos el mod *rewrite* de Apache y reiniciamos el servidor:

```
a2enmod rewrite
systemctl restart apache2
```

Al acceder mediante nuestro navegador web veremos la siguiente pantalla para configurar la utilidad:



[101]

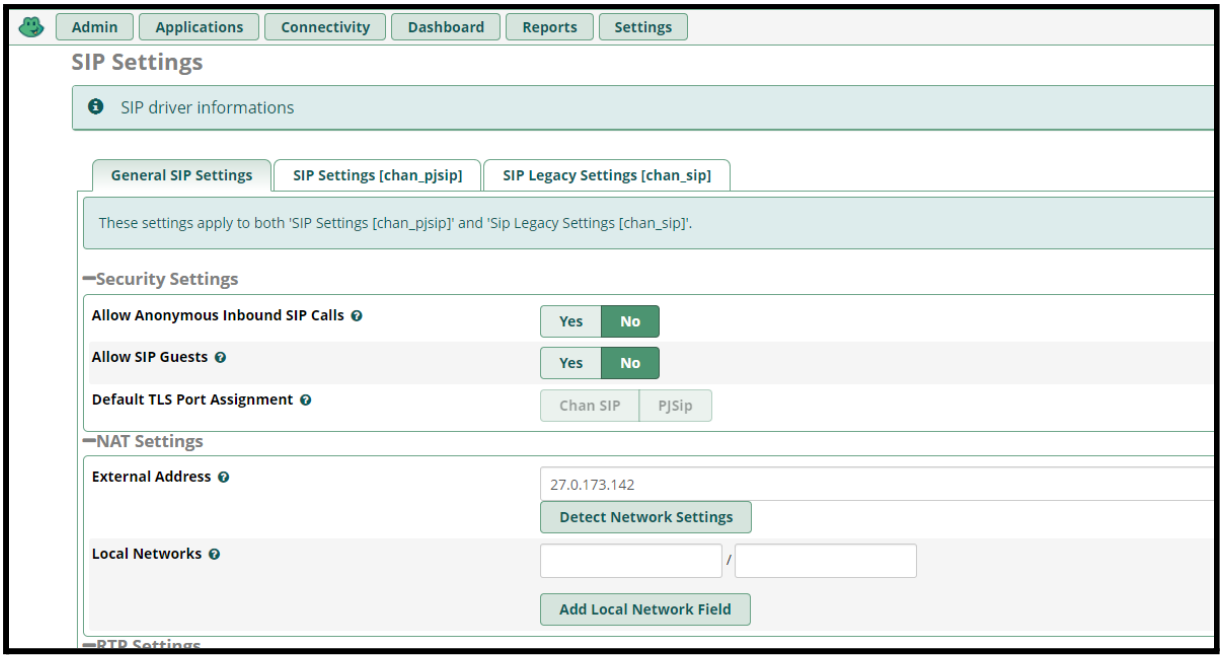
Indicamos los detalles necesarios para continuar. Aquí definiremos, entre otras opciones, el usuario administrador y la contraseña, que puede ser copiada desde el Anexo III de este documento.

5.4.2. Configuración vía web

5.4.2.1. Configuración SIP

Antes de nada, debemos evitar que clientes anónimos realicen llamadas a través de nuestro servidor. También debemos indicar nuestra dirección IP pública para la correcta gestión de las sesiones SIP. Podemos acceder desde la ruta *Settings* > *SIP Settings*.

[101] Esta imagen ha sido obtenida de un tutorial (nota al pie de página número 94), puesto que no se disponía de ella.



5.4.2.2. Activación de módulos FreePBX

Estos módulos son diferentes de los módulos activados durante la compilación de Asterisk, pero añaden funcionalidades interesantes para la gestión. En este servidor se encuentran activados los siguientes módulos:

Admin

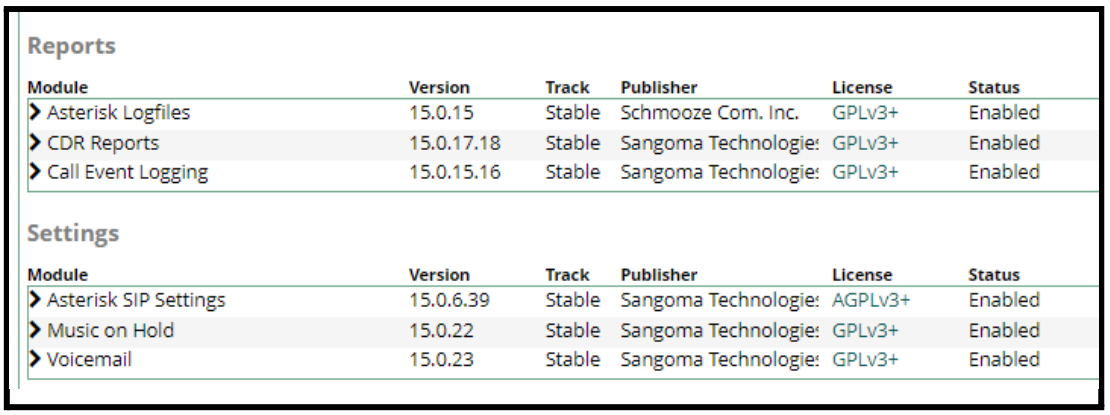
Module	Version	Track	Publisher	License	Status
Certificate Manager	15.0.49	Stable	Sangoma Technologie:	AGPLv3+	Enabled
Custom Applications	15.0.14	Stable	Sangoma Technologie:	GPLv3+	Enabled
Feature Code Admin	13.0.6.11	Stable	Sangoma Technologie:	GPLv3+	Enabled
FreePBX Framework	15.0.23	Stable	Sangoma Technologie:	GPLv2+	Enabled
Process Management	15.0.10	Stable	Sangoma Technologie:	AGPLv3+	Enabled
Recordings	15.0.3.16	Stable	Sangoma Technologie:	GPLv3+	Enabled
Sound Languages	15.0.5.10	Stable	Sangoma Technologie:	GPLv3+	Enabled
User Management	15.0.69.1	Stable	Sangoma Technologie:	AGPLv3+	Enabled

Applications

Module	Version	Track	Publisher	License	Status
Call Recording	15.0.7.24	Stable	Sangoma Technologie:	AGPLv3+	Enabled
Conferences	15.0.7.11	Stable	Sangoma Technologie:	GPLv3+	Enabled
Core	15.0.22	Stable	Sangoma Technologie:	GPLv3+	Enabled
IVR	15.0.29	Stable	Sangoma Technologie:	GPLv3+	Enabled
Info Services	15.0.3	Stable	Sangoma Technologie:	GPLv2+	Enabled
Set CallerID	15.0.9	Stable	Sangoma Technologie:	GPLv3+	Enabled

Dashboard

Module	Version	Track	Publisher	License	Status
System Dashboard	15.0.15	Stable	Sangoma Technologie:	AGPLv3+	Enabled



Reports

Module	Version	Track	Publisher	License	Status
Asterisk Logfiles	15.0.15	Stable	Schmooze Com. Inc.	GPLv3+	Enabled
CDR Reports	15.0.17.18	Stable	Sangoma Technologie:	GPLv3+	Enabled
Call Event Logging	15.0.15.16	Stable	Sangoma Technologie:	GPLv3+	Enabled

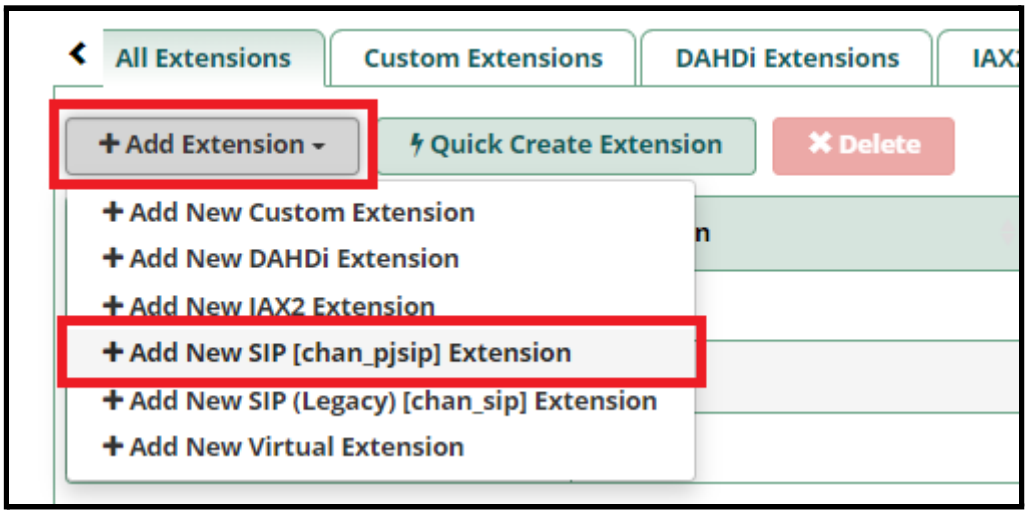
Settings

Module	Version	Track	Publisher	License	Status
Asterisk SIP Settings	15.0.6.39	Stable	Sangoma Technologie:	AGPLv3+	Enabled
Music on Hold	15.0.22	Stable	Sangoma Technologie:	GPLv3+	Enabled
Voicemail	15.0.23	Stable	Sangoma Technologie:	GPLv3+	Enabled

5.4.2.3. Creación de usuarios SIP

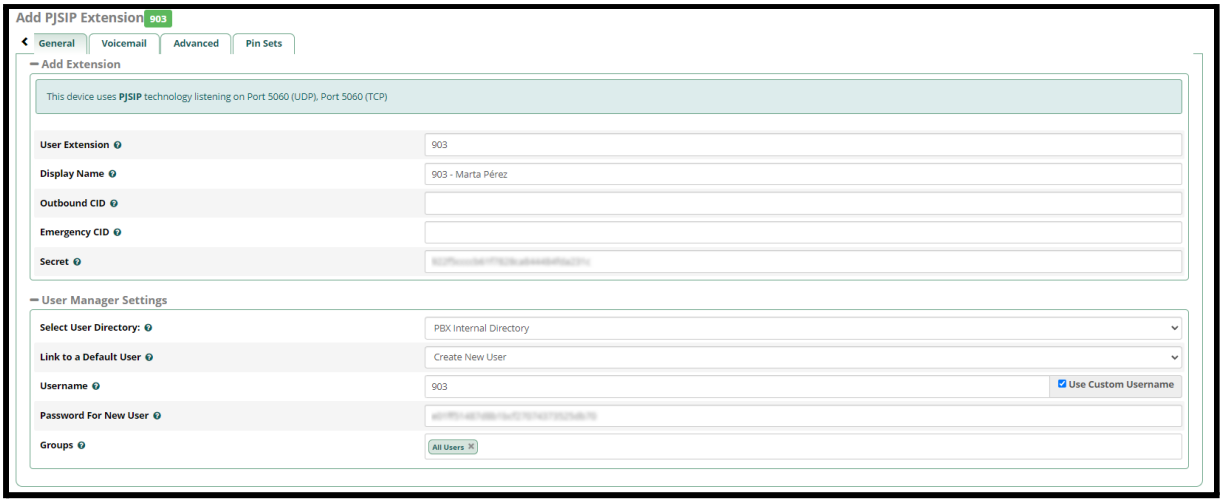
Para la creación de usuarios hay dos vías: crear el usuario, luego la extensión y por último enlazarlas; o crearlo todo al mismo momento.

Por comodidad, lo haremos todo en el mismo momento. Para ello, desde el menú superior nos desplazamos hasta *Applications* > *Extensions*. Aquí, hacemos clic en el botón *Add Extension*, para luego seleccionar el tipo *chan_pjsip*:



En la siguiente ventana, indicamos la extensión (será la 903 en este caso). También un nombre, para el que se ha elegido *903 - Marta Perez.*

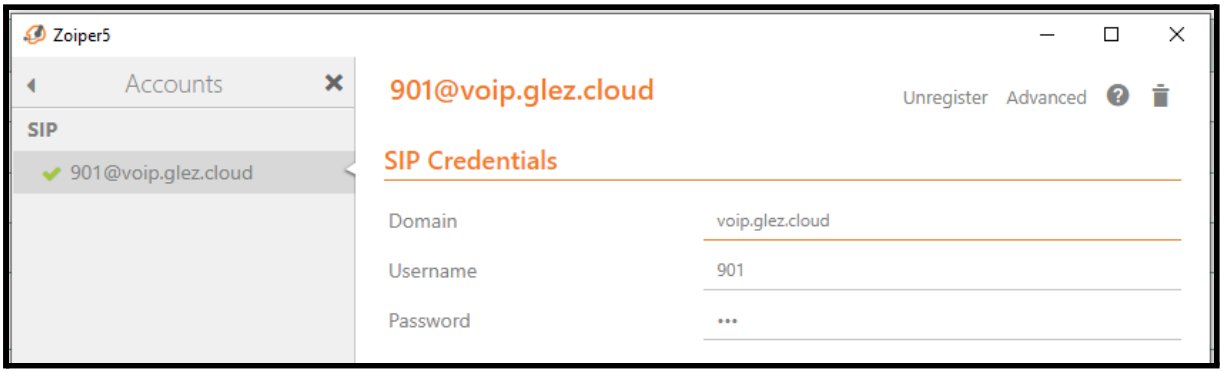
En la sección de *User Manager Settings*, seleccionamos que el usuario sea creado en el directorio interno del sistema. A su vez, marcamos la casilla de indicar un nombre de usuario personalizado, indicando también *903*. Copiamos la contraseña del usuario SIP.

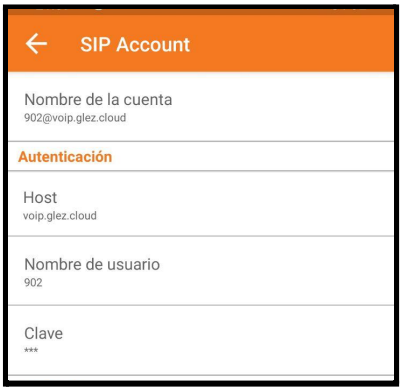


Se han creado extensiones para 3 usuarios: 901, 902 y 903. Las contraseñas quedarán reflejadas en el Anexo III de este documento.

Tanto en dispositivos móviles como en escritorio se ha utilizado Zoiper como softphone. No es la mejor aplicación, ni es software libre pero está bastante extendida.

Esta son las configuraciones de la aplicación de escritorio y de la aplicación móvil:

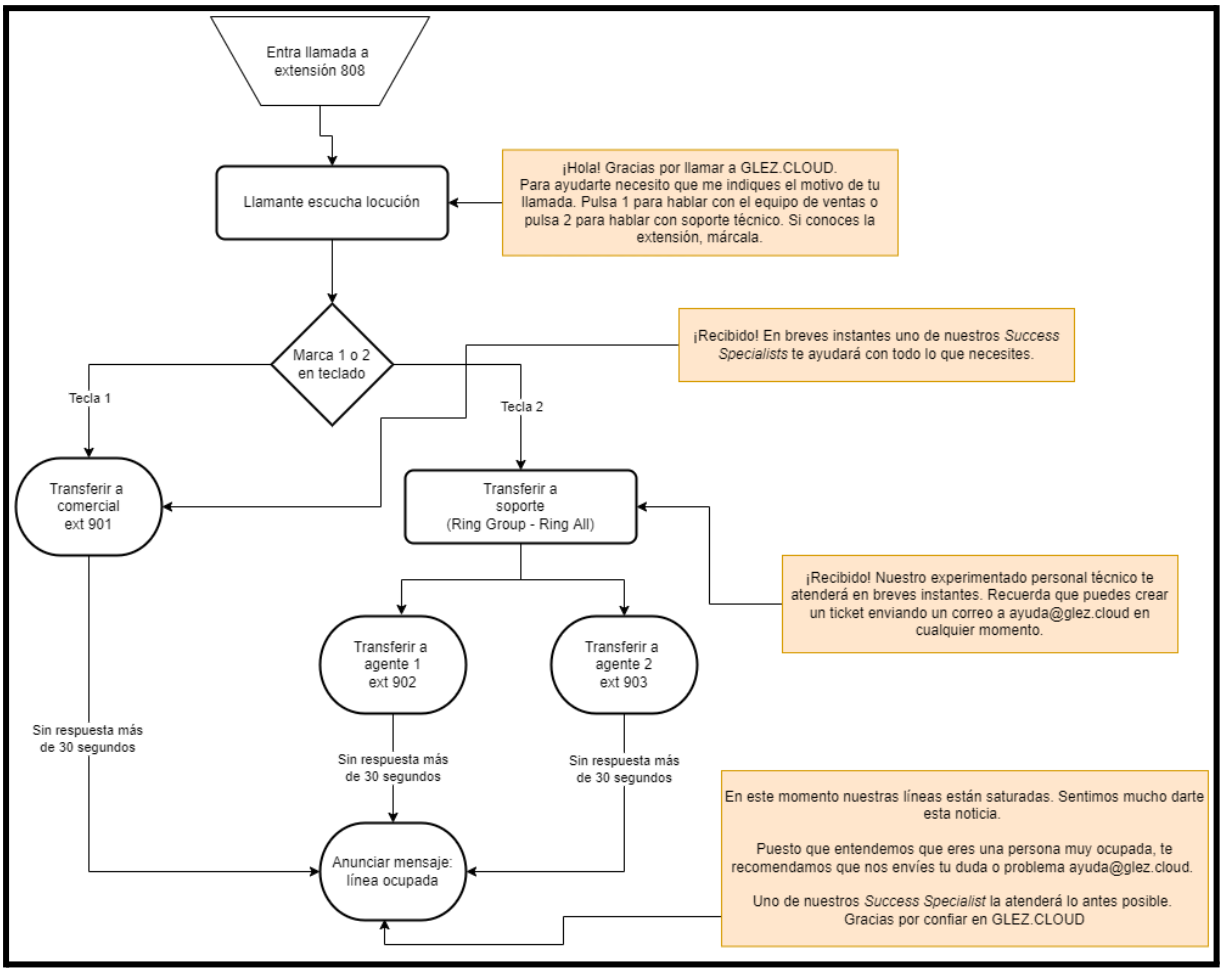




5.4.2.4. Creación de menú IVR

Diagrama del menú IVR

El objetivo de esta sección es generar en Asterisk/FreePBX la siguiente estructura IVR para recibir las llamadas de los clientes:

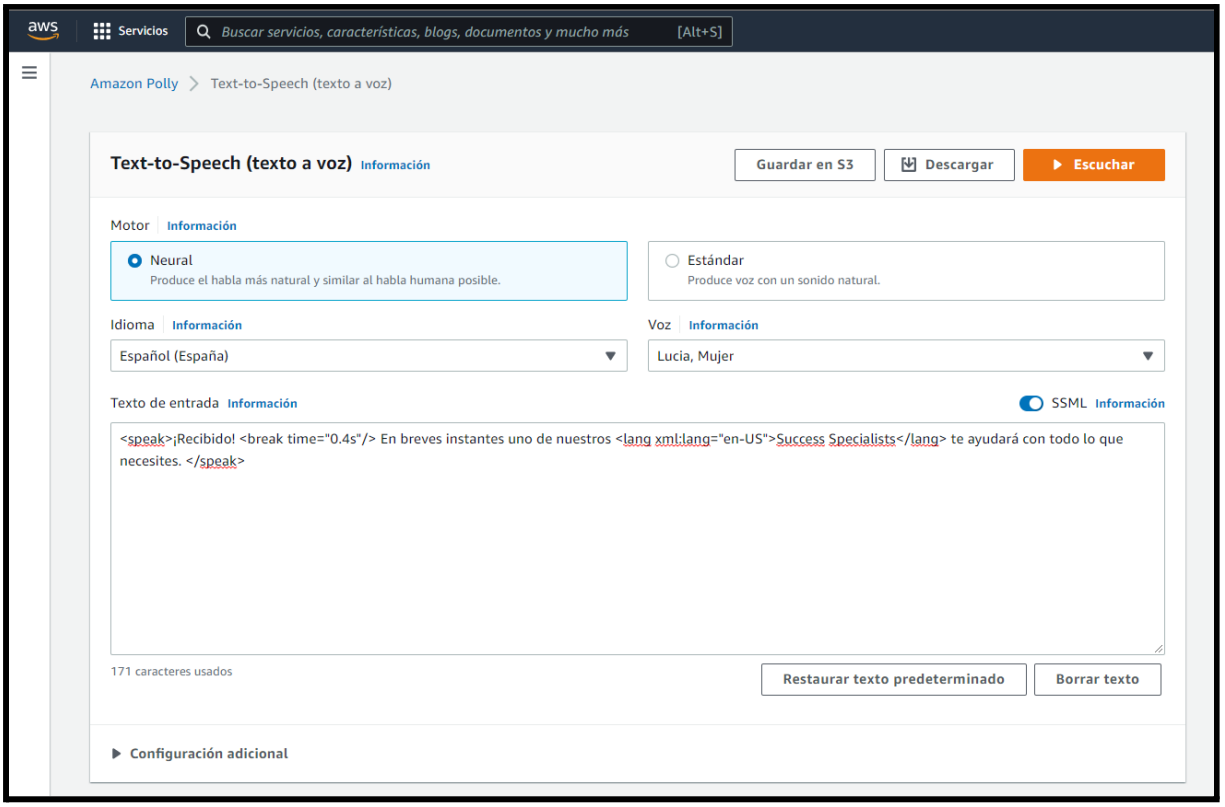


Generación de locuciones con Amazon Poly

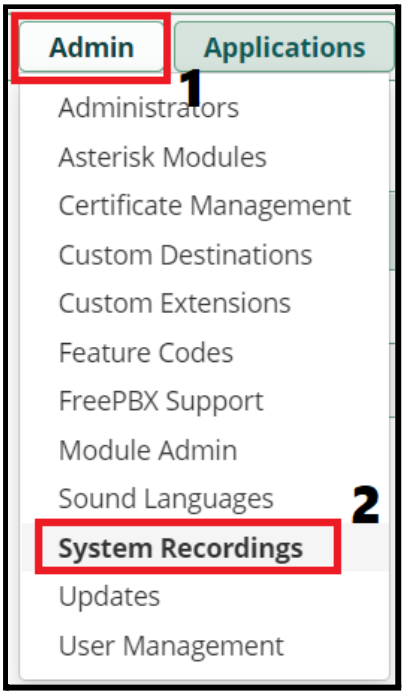
Usando esta herramienta se han generado los siguientes archivos de audio:

- *bienvenida.mp3* y *bienvenida.ogg*
- *busy.mp3* y *busy.ogg*
- *transfer2comercial.mp3* y *transfer2comercial.ogg*
- *transfer2tech.mp3* y *transfer2tech.ogg*

Aunque también se puede usar mediante API, la interfaz web es sencilla y suficiente para este nivel de uso:



Todos los audios están disponibles en el repositorio de GitHub[102]. Se han generado tanto en mp3 como en ogg puesto que el primero de los códecs es propietario y mucho más intensivo en CPU.



Subida de locuciones a Asterisk

Antes de poder disponer de las locuciones en Asterisk debemos subirlas al sistema.

Aunque los nombres de las mismas son bastante descriptivos, indicaremos en este momento el uso de cada grabación.

- *bienvenida.[ogg|mp3]* es la que el llamante escucha y le indica las opciones del menú.
- transfer2[comercia|tech].[ogg.mp3] son las grabaciones que escucha la persona llamante a la hora de ser transferidos al departamento

[102] https://github.com/gonzaleztroyano/ASIR2-PFC/tree/main/5-support-voip/voip_locuciones

comercial o al departamento de soporte técnico.

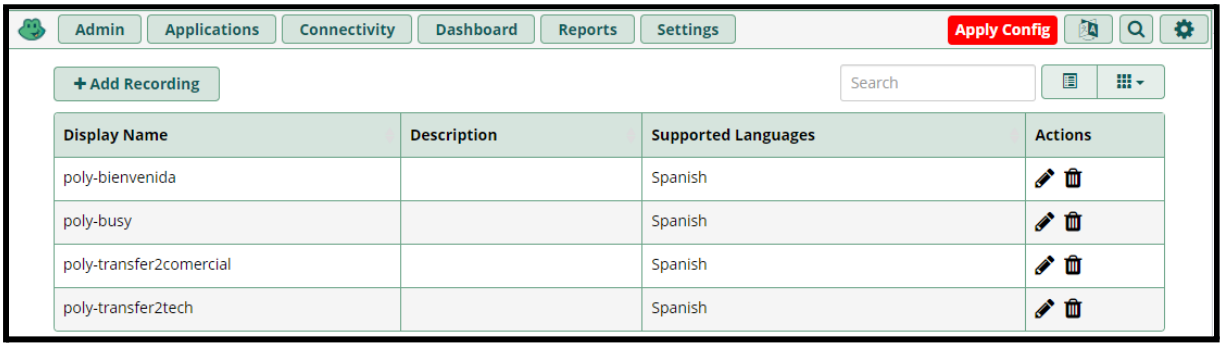
- busy.[ogg|mp3] es la grabación que se reproducirá en el caso de que ningún agente haya atendido la llamada.

Haciendo clic en *Admin* > *System Recordings* veremos un botón en el que se puede leer *Add Recording*. Pulsamos para añadir una nueva grabación.

Indicamos un nombre. En nuestro caso todas las grabaciones se llamarán *poly-* seguido del objetivo de la grabación. *poly-busy* para el mensaje de línea ocupada y *poly-bienvenida* para las instrucciones iniciales del IVR, por ejemplo.

Subiremos desde nuestro equipo las grabaciones y seleccionaremos convertir a wav el archivo de audio.

Una vez subidas todas las grabaciones, hacemos clic en el botón *Apply Config* de la parte superior derecha de la pantalla:



Creación de selector principal

Para la creación del selector principal, en la que el usuario o la usuaria llamante decide si desea ponerse en contacto con *Atención Comercial* o *Soporte Técnico* debemos realizar dos tareas:

- Crear un menú IVR
- Crear una extensión virtual para “invocar” el menú IVR creado.

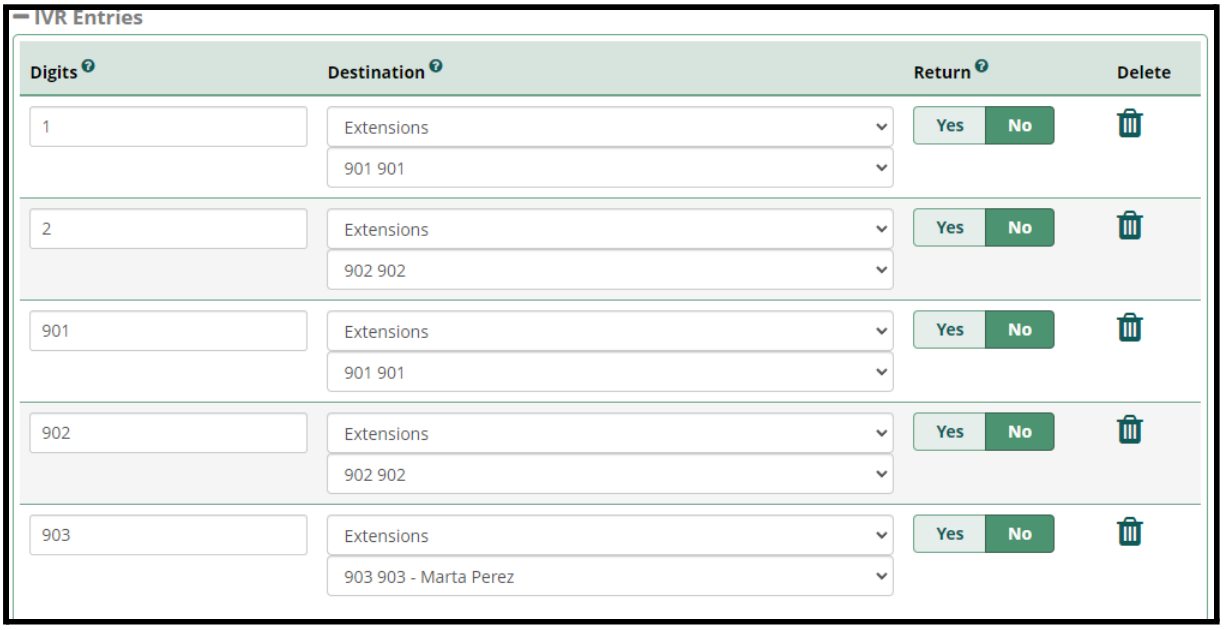
Para la creación de un menú IVR basta con navegar hasta *Applications* > *IVR*. Si no lo viéramos aquí disponible, sería necesaria la instalación del módulo desde *Admin* > *Module Admin*.

Hacemos clic en *Add IVR*. Aquí rellenamos los datos como el nombre (indicando *selector*). En announcement seleccionamos la grabación que hemos añadido antes, *poly-bienvenida*.

El resto de ajustes los podemos dejar por defecto, aunque dependerá un poco de nuestro gusto en lo relativo a mensajes de error y tiempo a esperar para dar por terminada la llamada.

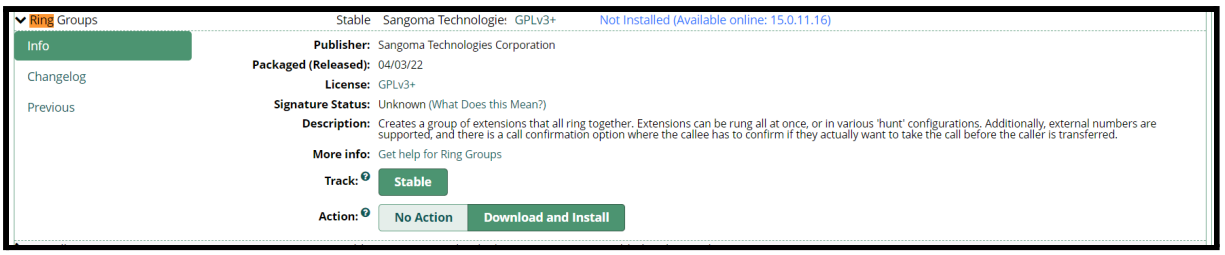
En la parte inferior de la pantalla podremos ver los destinos a los que el menú IVR enviará a los usuarios según las teclas pulsadas.

En este momento al pulsar la tecla “2” se envía al usuario a la extensión 902, puesto que no está generado todavía el *Ring Group* de Soporte. Cuando se cree, se cambiará a aquí la configuración para llamar a este *Ring Group* en conjunto y no solo a uno de los usuarios que lo forman (recordemos que las extensiones 902 y 903 son miembros de este grupo). Esto es lo que vemos:

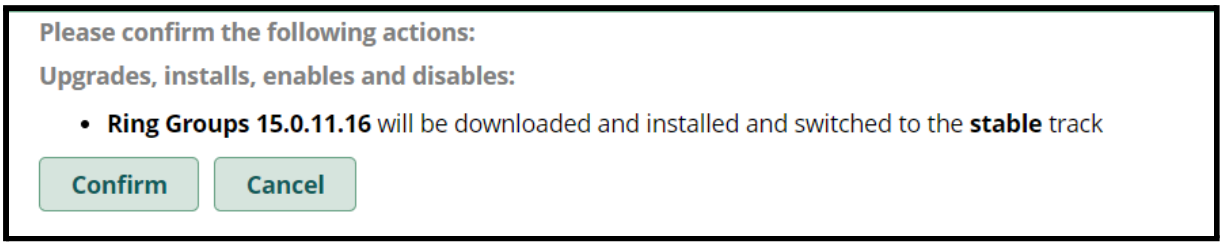


Creación de grupo “Soporte”

Antes de nada, procedemos a instalar el siguiente módulo de FreePBX. Los módulos de FreePBX son diferentes a los módulos de asterisk y añaden funcionalidades extra.



Cuando nos sea solicitado por el asistente de instalación, confirmamos los cambios:



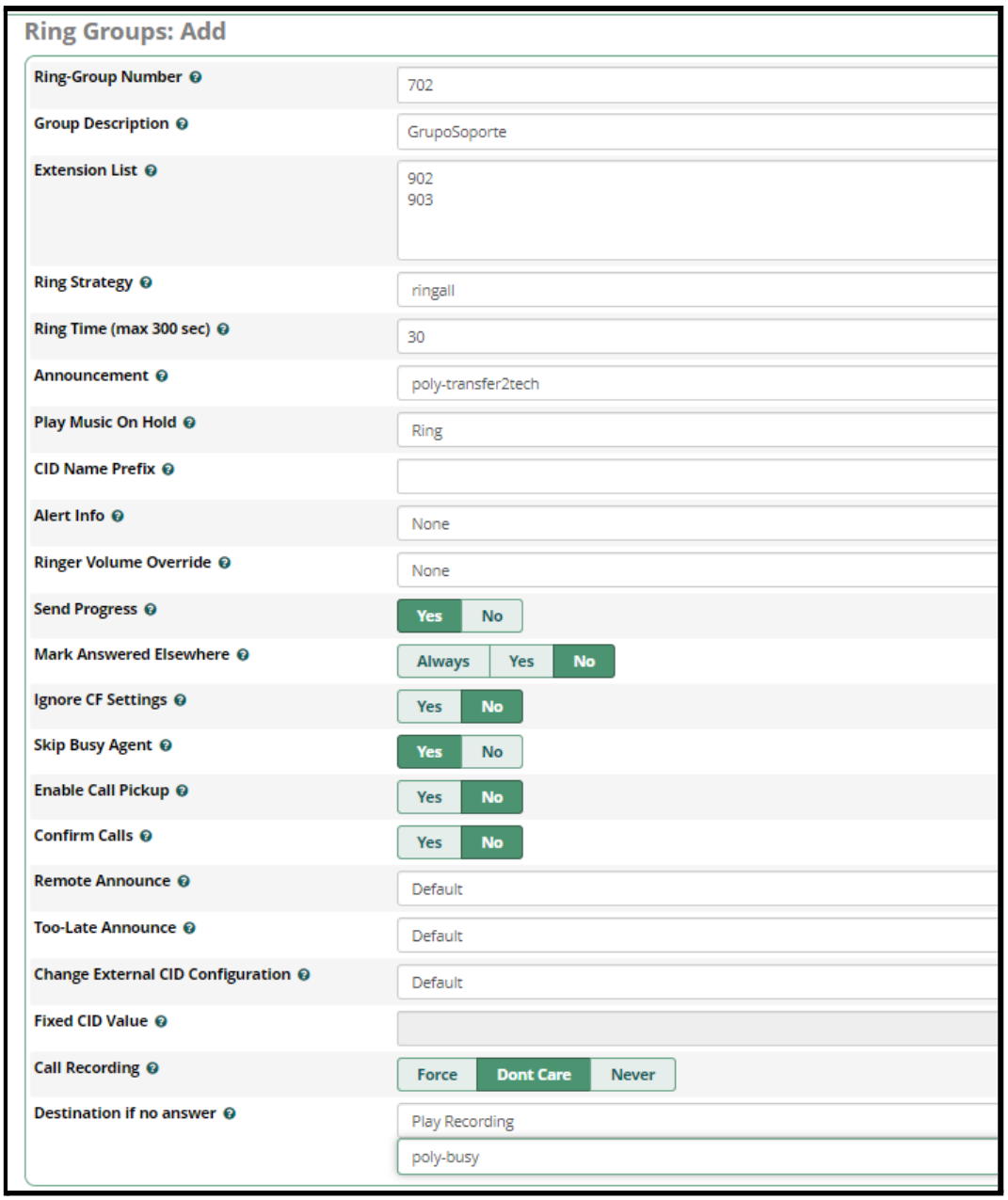
En este momento ya vemos disponibles los *Ring Groups* dentro del menú *Applications*. Hacemos clic sobre esta opción para después seleccionar *Add Ring Group*.

En *Ring Group Number* indicamos “702” y definimos como descripción una descriptiva, *GrupoSoporte* en nuestro caso. En las extensiones, seleccionamos 902 y 903 como extensiones destino. En *Ring Strategy*, seleccionamos *ringall* para que llame a todos al mismo tiempo.

Seleccionamos 30 segundos como tiempo máximo de espera y la locución “poly-transfer2tech” que hemos creado anteriormente. Seleccionamos que el sistema no llame a un usuario que esté en otra llamada y que reproduzca el mensaje de ocupado si ningún agente contesta.

El resto de ajustes los podemos mantener por defecto, o bien modificarlos si lo consideramos necesario. En cualquier caso, es importante estar seguros de los cambios que estamos introduciendo, pues podemos dejar el sistema fuera de funcionamiento por un ajuste “pequeño”.

Así queda la configuración:



En este momento, editamos también el menú IVR inicial para que al pulsar 2 el usuario llamante sea redirigido al Ring Group. El IVR queda de la siguiente manera:

6. Soluciones *Out-of-the-box*

En esta sección del proyecto, se dispondrá de un servicio de gestión creado por otra empresa. En este caso se ha optado por el sistema de gestión Plesk[103], al ser junto con cPanel, el más utilizado a nivel global por empresas de servicios de IT como la que pretendemos emular.

Para esta sección del proyecto se ha alojado el servidor en axarnet.es, que ha colaborado con el proyecto patrocinando los servidores utilizados.

Una vez el proveedor, en nuestro caso axarnet.es, ha desplegado el servidor e instalado en sistema de gestión (Plesk) debemos comenzar la configuración.

Una de las ventajas de trabajar con axarnet es que su equipo técnico especializado está para todo lo que necesitemos y son ellos/as quienes ejecutan la instalación. En el caso de que decidamos contratar el servidor en un proveedor que no ofrezca esta facilidad, podemos obtener la licencia de Plesk a través de su página web[104,105] y seguir las instrucciones de instalación[106].

[103] https://www.plesk.com/
[104] https://www.plesk.com/pricing/
[105] https://scdn1.plesk.com/wp-content/uploads/2020/09/29113042/plesk_licensing_guide_online-customers.pdf
[106] https://docs.plesk.com/en-US/obsidian/advanced-administration-guide-linux/about-this-guide.68553/

6.1. Inicialización de la configuración

6.1.1. Inicio de sesión

Una vez hemos iniciado sesión mediante SSH, debemos ejecutar el siguiente comando para obtener un enlace de inicio de sesión único con el que podremos comenzar la configuración:

```
root@glez-cloud.vservers.es:~ # plesk login
https://glez-cloud.vservers.es:8443/login?secret=8NSCL
```

Accediendo con este enlace, automáticamente iniciamos sesión como usuario administrador. También hemos procedido a cambiar la contraseña de root, que se encuentra disponible en el Anexo III de este documento como referencia.

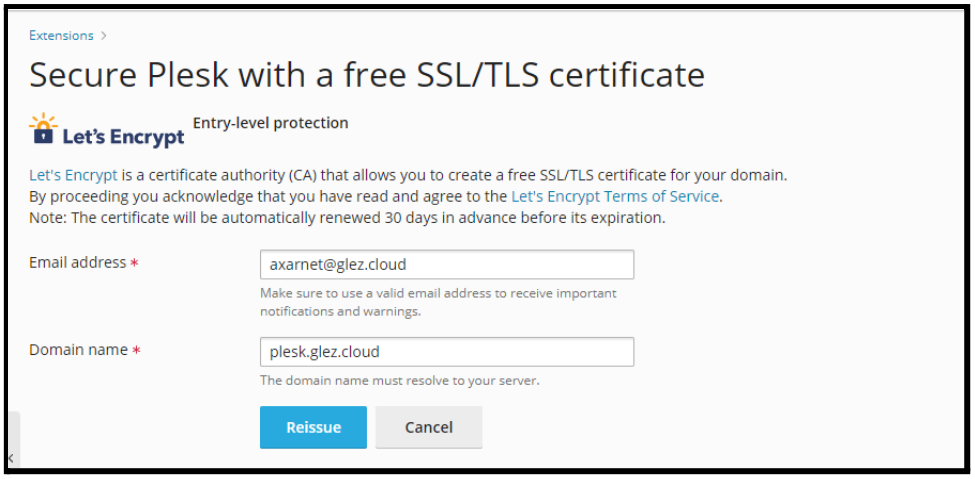
6.1.2. Cambio de hostname y generación de certificado

Cambiaremos el hostname de plesk desde su página de ajustes. Para ello, accedemos a *Tools & Settings* > *General Settings* > *Server Settings.* Aquí, basta con cambiar el hostname en el primer campo dedicado a este fin. En este caso se ha decidido asignar el siguiente nombre de dominio:

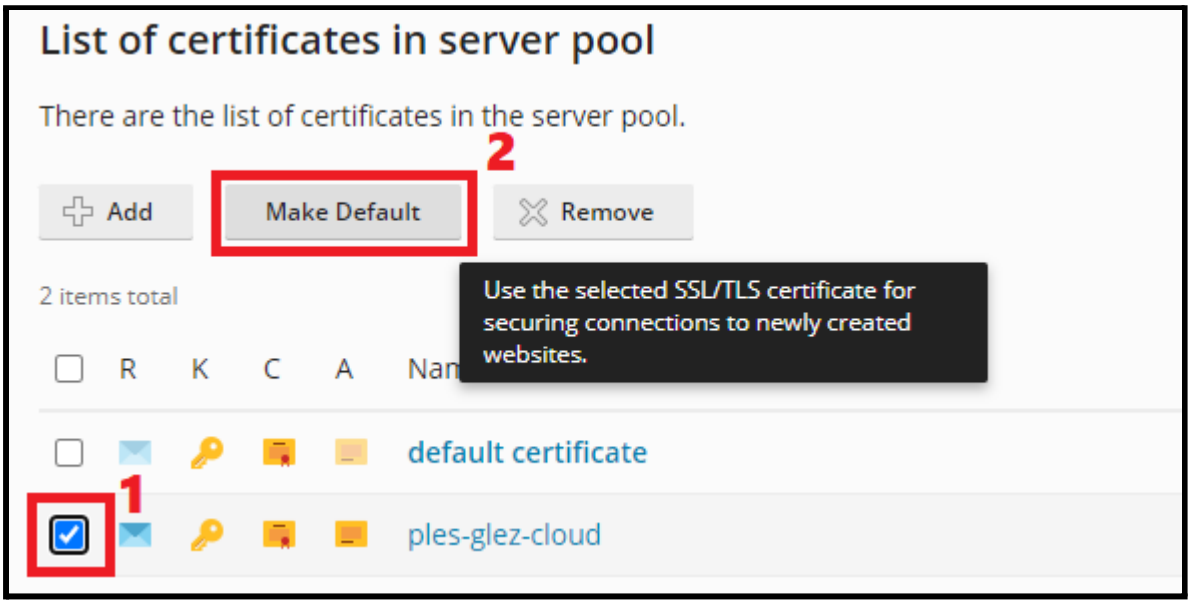
```
plesk.glez.cloud
```

Para mantener la coherencia entre sistemas, desde el backoffice de axarnet, editamos el nombre del servidor:

Una vez hecho esto, generaremos el certificado TLS con la integración nativa que tiene el sistema con Let 's Encrypt.

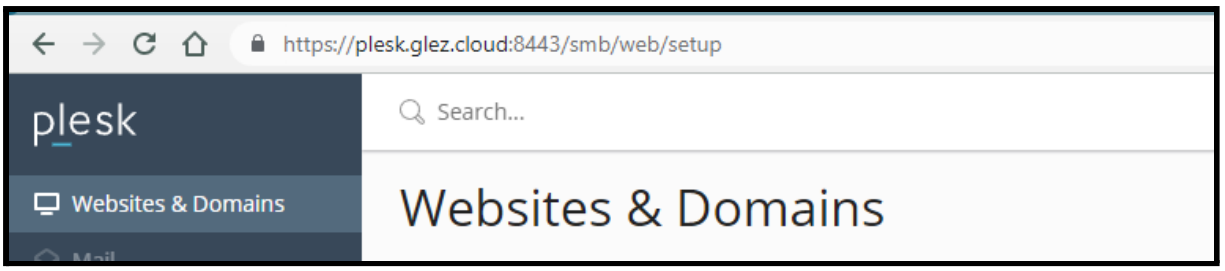


Una vez generado el certificado, lo renombraremos para localizarlo más fácilmente. También lo definiremos como certificado predeterminado:



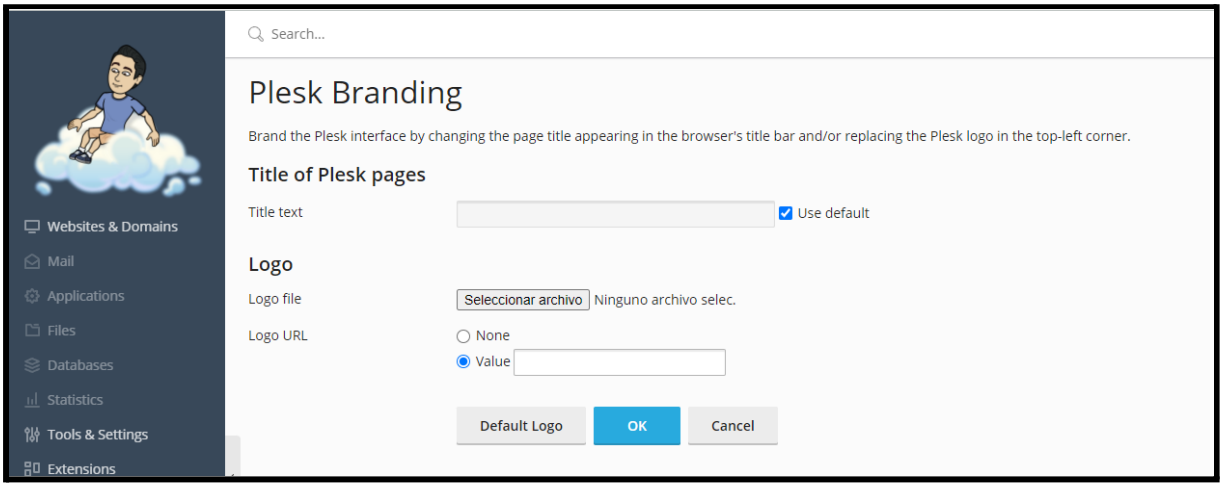
Editaremos la URL del navegador para indicar el nuevo dominio y recargamos la página. Puesto que las cookies no se comparten entre dominios (y antes estábamos usando un subdominio de vservers.es) debemos iniciar la sesión de nuevo.

Al hacerlo, la veremos el panel de administración de Plesk y el “candadito” de HTTPS:



6.1.3 Cambio de branding

Como se puede ver en la imagen anterior, en la parte superior izquierda del menú se puede ver el logo de Plesk. Subiremos el logo de GLEZCLOUD para adaptarlo a nuestra imagen corporativa. Una vez hecho, el resultado lo veremos inmediatamente:

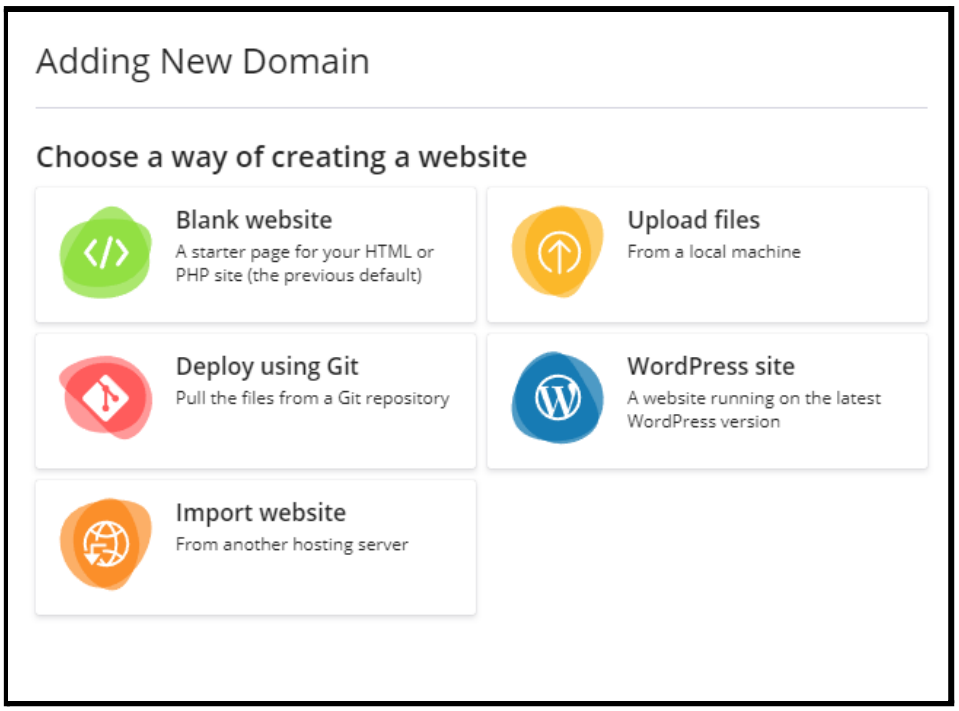


6.2. Adición de un nuevo dominio

6.2.1. Dar de alta el sitio web en Plesk

En este momento, añadiremos el dominio “ahorramás.es” a Plesk. Para hacerlo, nos dirigimos a la página principal de Plesk. Hacemos clic en “+ Add domain”.

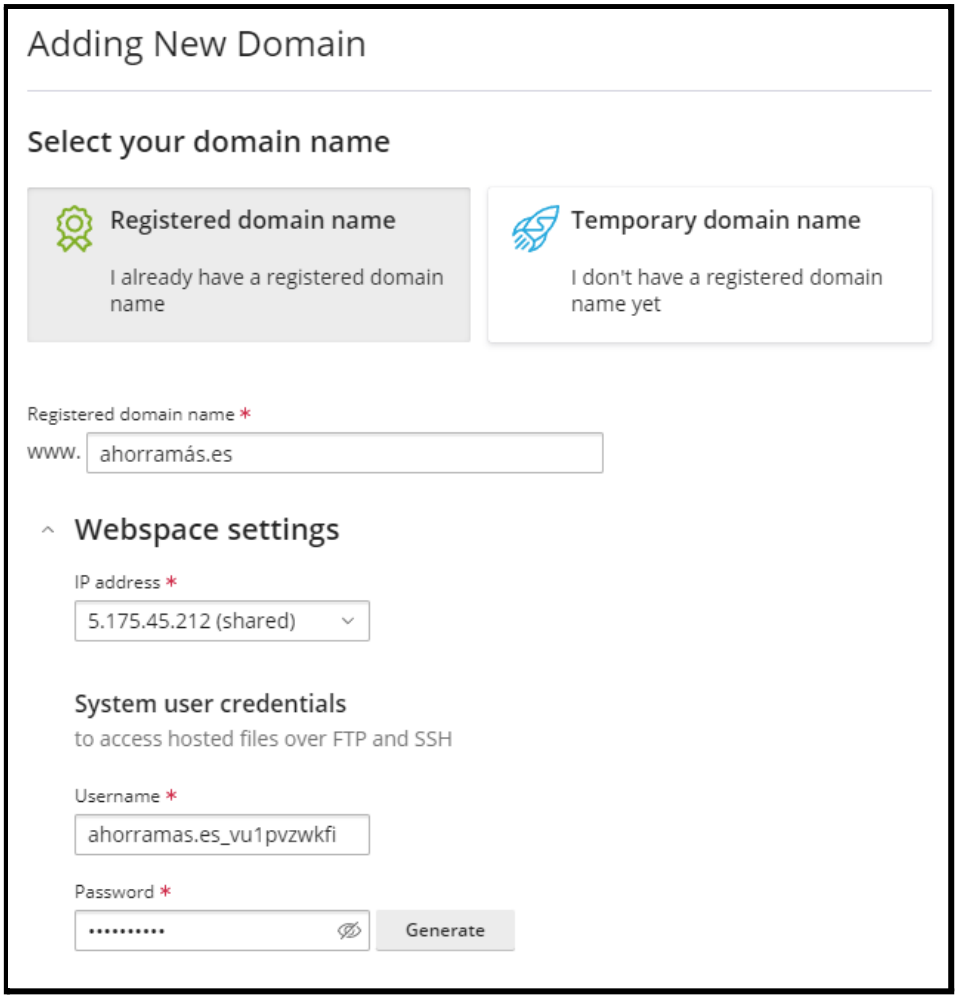
Al hacerlo, se nos despliega el panel de opciones:



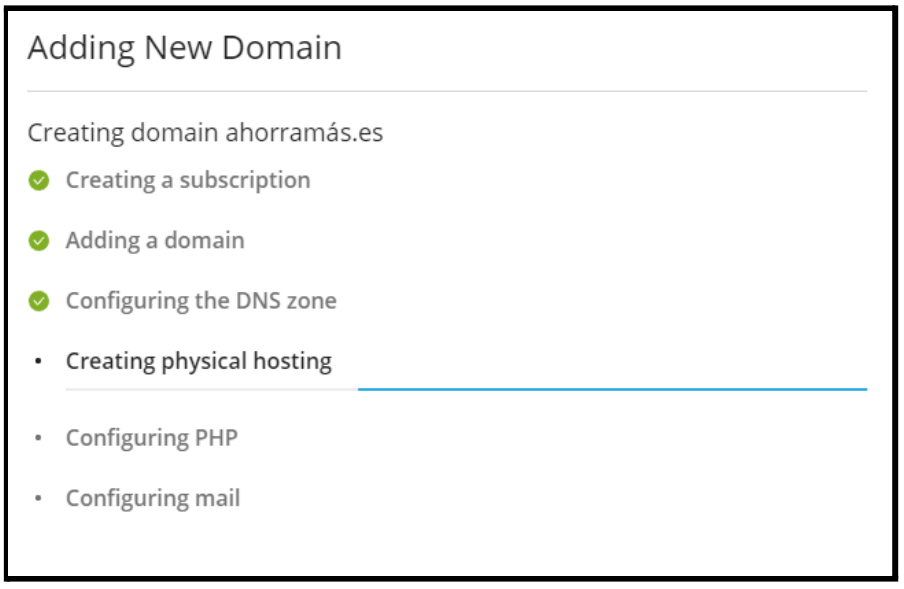
Elegimos “Upload files”. El archivo HTML que subiremos es realmente básico, pero como demostración es suficiente. Es el siguiente archivo:

```
<html>
	<head>
		<title>AHORRAMÁS NO ES</title>
	</head>
	<body>
		<p>Lamentablemente, esto no es Ahorramás. Puedes acceder desde <a
href="https://www.ahorramas.com/">aquí</a>
	</body>
</html>
```

Al seleccionar la creación de un nuevo sitio mediante la subida de archivos (o cualquier otra opción) nos aparecerá la siguiente pantalla para indicar el dominio a añadir y los detalles del usuario



En la parte inferior de la página veremos la opción para aceptar los cambios. Acto seguido, el sistema aplicará los cambios pertinentes.

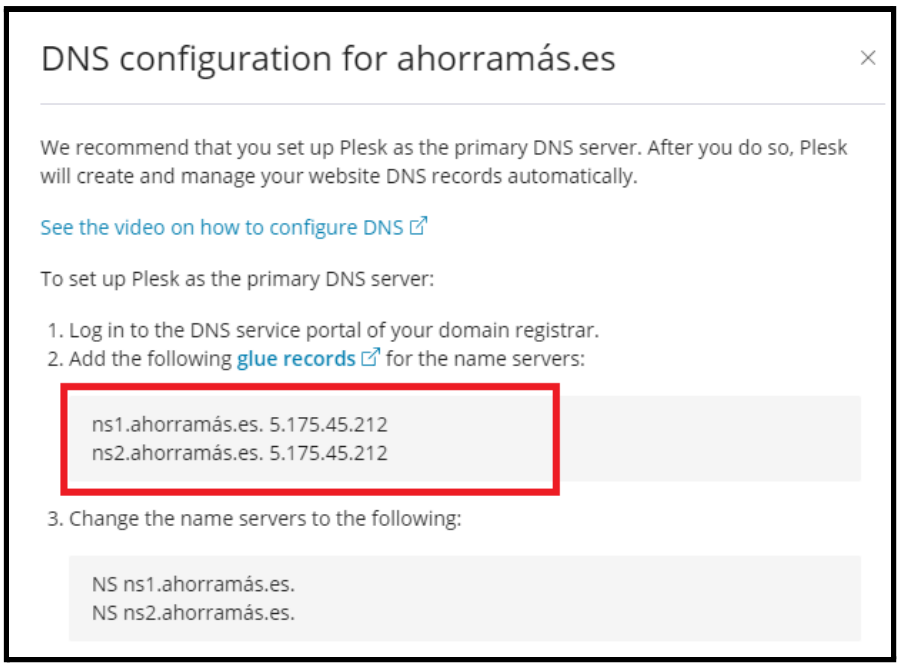


En la siguiente pantalla, subiremos el archivo HTML mostrado anteriormente.

6.2.2. Resolución DNS del dominio

Puesto que el servidor Plesk no es el servidor DNS autoritativo para la zona ahorramás.es, veremos un error:

El propio servidor Plesk nos muestra los valores que tenemos que indicar en el registrador del dominio:

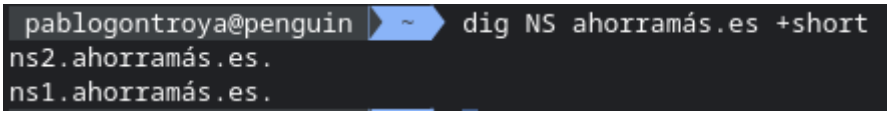


En dinahosting, primero crearemos las dos zonas que servirán de registros “glue”. En este caso, ns1 y ns2. El proceso es sencillo.

Los cambios en los servidores DNS principales de un dominio pueden tardar varias horas pues no dependen únicamente de nuestro lado, sino del registro central del TLD, NIC.es en el caso del TLD para España.

Podemos comprobarlo ejecutando el siguiente comando:

```
dig NS ahorramás.es +short
```



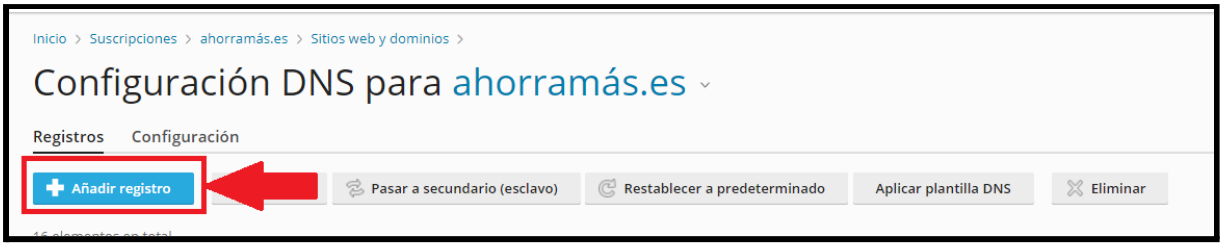
Resulta realmente útil utilizar los modificadores del comando `dig`, como `+trace`. De esta forma podemos ver toda la resolución (desde los servidores raíz hasta el autoritativo):

```
dig NS ahorramás.es +trace +nodnssec
```

Se añade `+nodnssec` para evitar que consulte (y muestre a través de la terminal) la verificación de DNSSEC. El resultado es el siguiente:

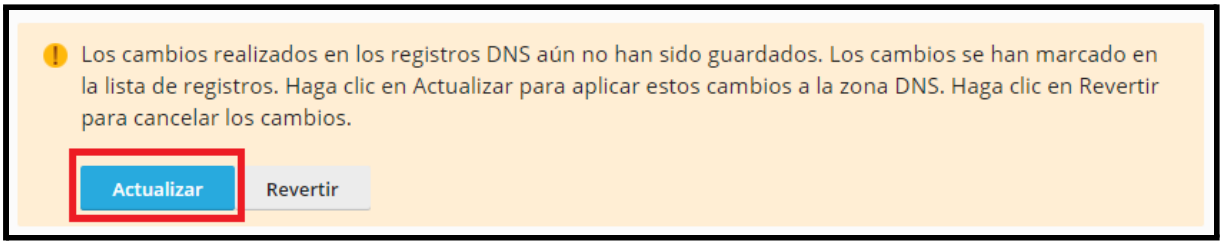
```
; <<>> DiG 9.11.5-P4-5.1+deb10u7-Debian <<>> NS ahorramás.es +trace +nodnssec
;; global options: +cmd
.                       512778  IN      NS      a.root-servers.net.
.                       512778  IN      NS      b.root-servers.net.
.                       512778  IN      NS      c.root-servers.net.
.                       512778  IN      NS      d.root-servers.net.
.                       512778  IN      NS      e.root-servers.net.
.                       512778  IN      NS      f.root-servers.net.
.                       512778  IN      NS      g.root-servers.net.
.                       512778  IN      NS      h.root-servers.net.
.                       512778  IN      NS      i.root-servers.net.
.                       512778  IN      NS      j.root-servers.net.
.                       512778  IN      NS      k.root-servers.net.
.                       512778  IN      NS      l.root-servers.net.
.                       512778  IN      NS      m.root-servers.net.
;; Received 811 bytes from 100.115.92.193#53(100.115.92.193) in 23 ms

es.                     172800  IN      NS      a.nic.es.
es.                     172800  IN      NS      c.nic.es.
es.                     172800  IN      NS      g.nic.es.
es.                     172800  IN      NS      h.nic.es.
;; Received 292 bytes from 193.0.14.129#53(k.root-servers.net) in 7 ms

ahorramás.es.           86400   IN      NS      plesk.glez.cloud.
ahorramás.es.           86400   IN      NS      plesk-2.glez.cloud.
;; Received 128 bytes from 194.69.254.1#53(a.nic.es) in 41 ms

ahorramás.es.           86400   IN      NS      ns2.ahorramás.es.
ahorramás.es.           86400   IN      NS      ns1.ahorramás.es.
;; Received 144 bytes from 5.175.45.212#53(plesk-2.glez.cloud) in 11 ms
```

Si en este momento añadimos un nuevo registro en el panel de control de Plesk, usando el botón creado para este fin:



Vemos que la interfaz es realmente sencilla:

Es especialmente importante guardar los cambios, para que estos se apliquen a nivel DNS y sean públicos:



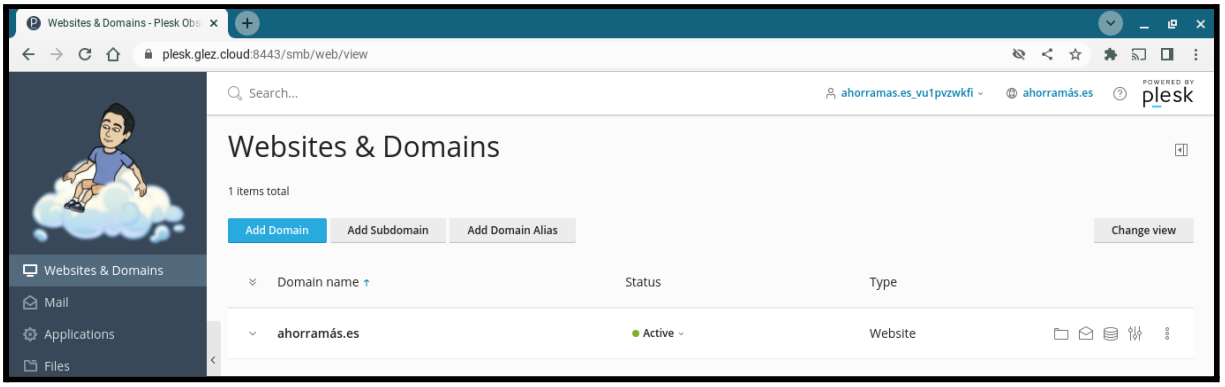
Si en este momento realizamos una consulta DNS podemos comprobar la correcta aplicación de los cambios:

```
dig TXT ahorramás.es +short
```

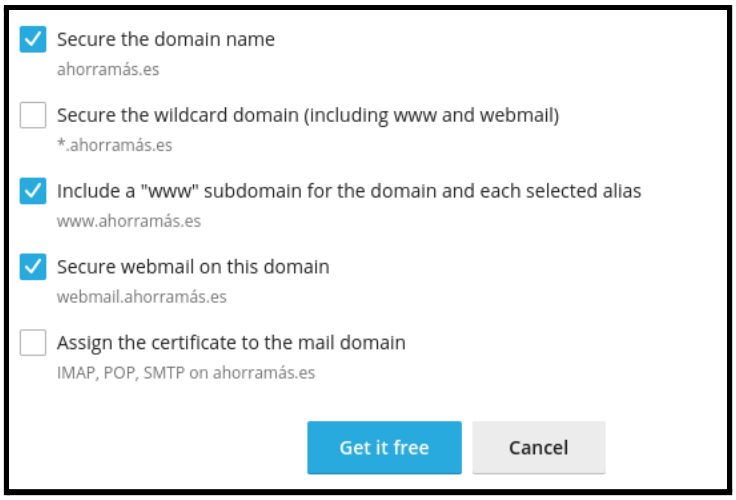
```
"Funcionando en Plesk"
"v=spf1 +a +mx +a:plesk.glez.cloud -all"
```

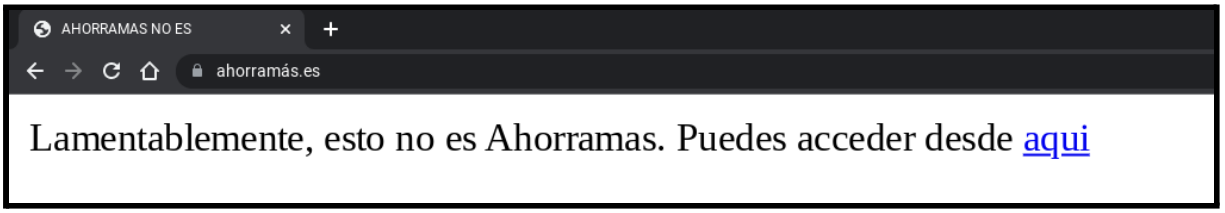
6.2.3. Panel de “cliente”

En este apartado iniciaremos sesión con el usuario cliente. La interfaz es realmente similar a la usada por el super administrador:



Haremos clic en *SSL/TLS certificates* para generar los certificados correspondientes, pues en este momento al acceder a *ahorrmás.es* el servidor nos envía el certificado de *plesk.glez.cloud*. Puesto que posteriormente probaremos el webmail, generaremos también certificados para estos subdominios.





El cliente desde su panel tiene la capacidad de realizar multitud de funciones, entre ellas podemos destacar:

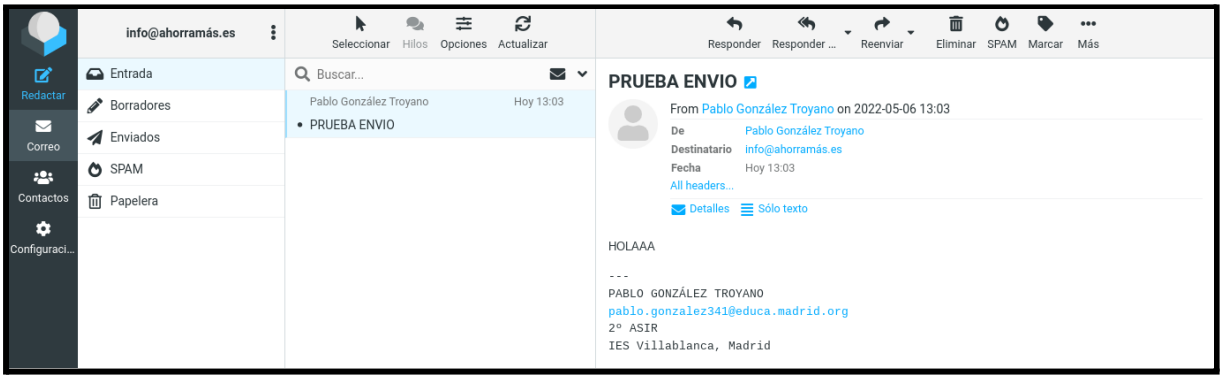
- Añadir dominios a su plan de hosting, dependiendo de su plan.
- Gestionar los ajustes de sus sitios web a nivel general.
- Acceder a archivos y logs. Mediante la propia interfaz web o usando un cliente FTP, previa activación desde Plesk.
- Crear bases de datos y gestionarlas. Tiene a su disposición phpMyAdmin para la gestión de las bases de datos mediante un navegador web.
- Crear y gestionar las cuentas y direcciones de correo electrónico, así como leer los mensajes usando el webmail, que posteriormente comentaremos.
- Editar las entradas DNS de sus dominios.
- Instalar aplicaciones de forma sencilla, como Joomla, WordPress y Drupal.
- Visualizar estadísticas de tráfico y uso de almacenamiento.
- Añadir sub-usuarios de gestión a su cuenta en Plesk.

Todas estas opciones dependen del plan de servicio y/o suscripción aplicable para el cliente en Plesk. Hasta este momento no se ha utilizado esta función, que trataremos con más detalle en un apartado a continuación.

6.3. Uso de webmail

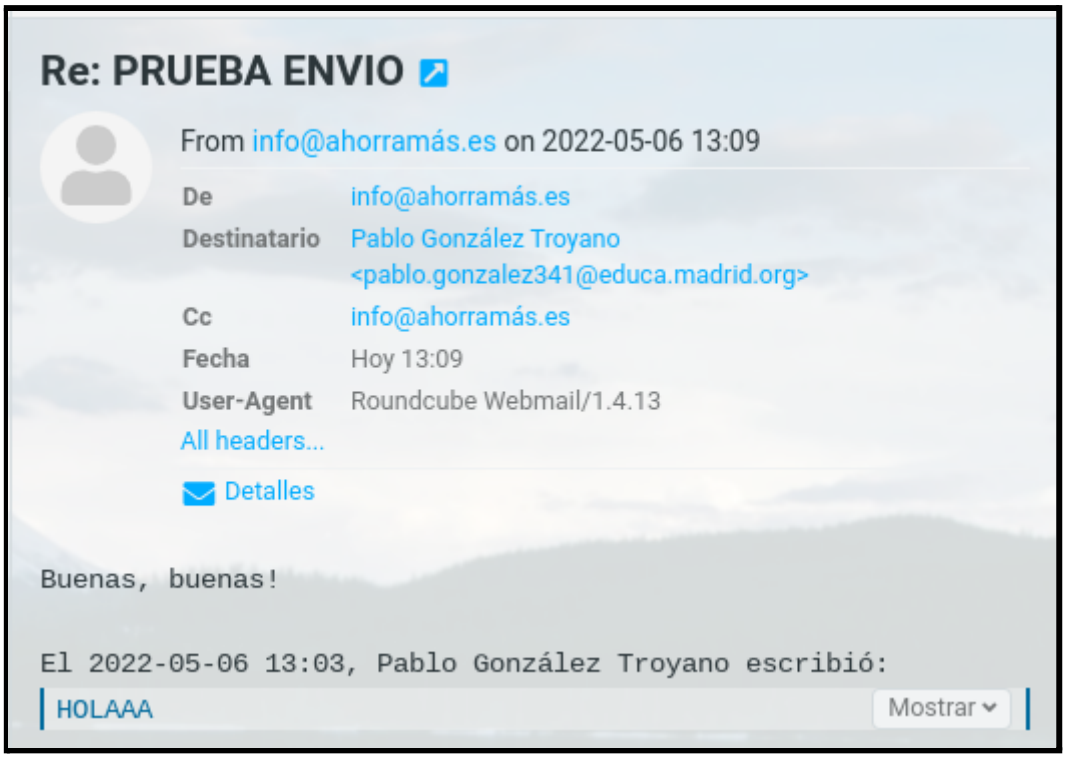
El webmail está basado en *Roundcube*, aunque es posible usar cualquier otro que deseemos. *Roundcube* es el mismo webmail que utiliza la Comunidad de Madrid para correoweb.educa.madrid.org.

Se ha enviado un correo electrónico desde la cuenta del Instituto y se comprueba como se recibe correctamente:



Para comprobar el envío de correo electrónico desde Plesk, se responde al mensaje.

El menaje se ha recibido correctamente:

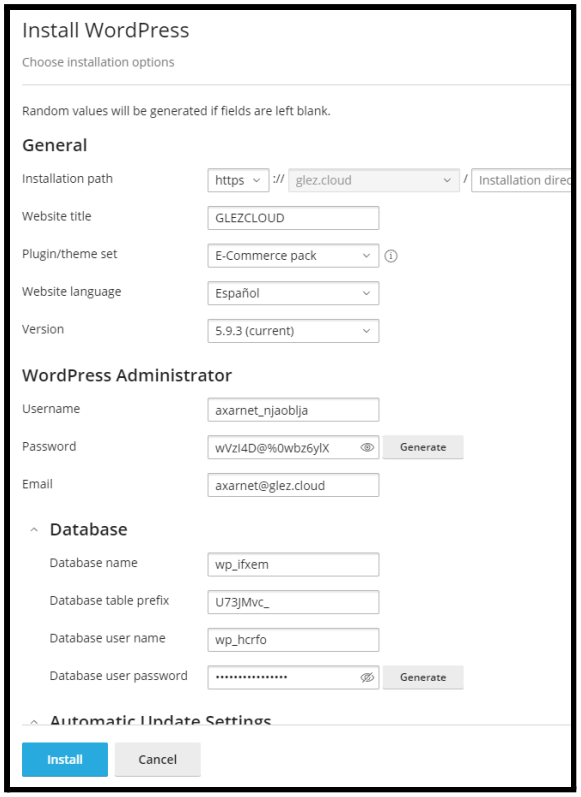


Las cabeceras, tanto del mensaje enviado[107] hacia Plesk, como este otro mensaje enviado[108] desde Plesk se han publicado en GitHub y se encuentran disponibles en el Anexo VIII de este documento.

[107] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/4-plesk/correo-educa2Plesk.txt
[108] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/4-plesk/correo-Plesk2Educa.txt

6.4. WordPress para la empresa

6.4.1. Instalación de WordPress

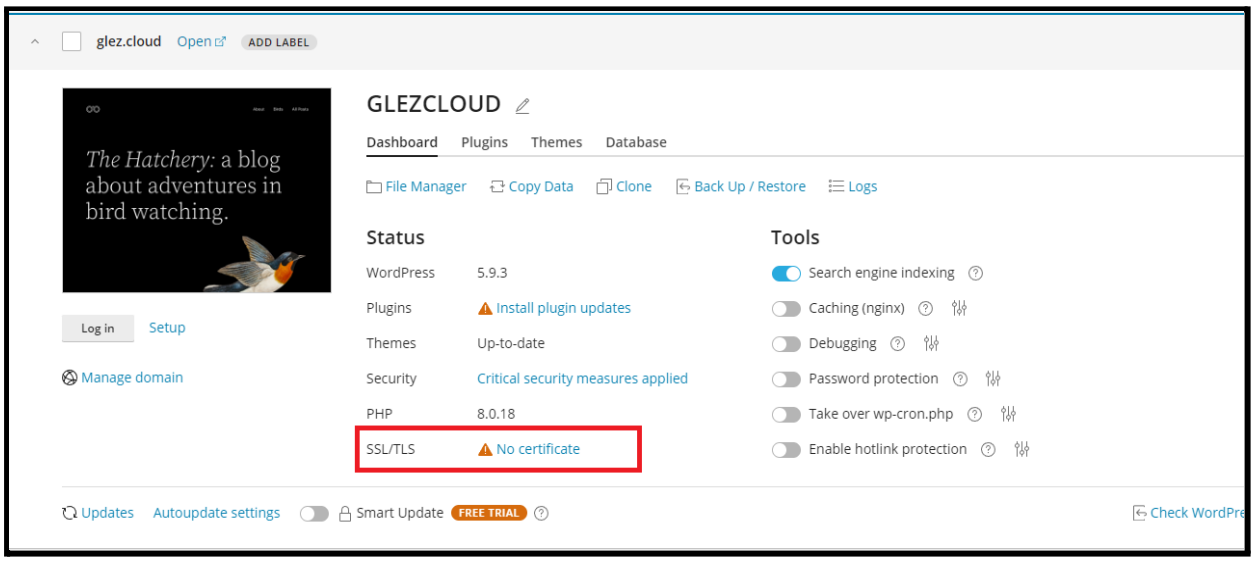


Seguimos un proceso similar al anterior, pero ahora indicamos que queremos crear un sitio web en WordPress. El sistema nos pregunta una serie de datos para crear y configurar el sitio, tal y como se puede ver en la imagen a la izquierda de este texto.

Una vez terminado el proceso, que puede tardar varios minutos, tendremos instalado nuestro sitio en WordPress. Actualizaremos nuestro registro A para que glez.cloud apunte a 5.175.45.212.

6.4.2. Instalación del certificado TLS/SSL

Como podemos ver en el resumen del sitio web, no dispone de certificado:



Al hacer clic sobre este aviso, nos saltará un bocadillo con la opción *Get an SSL/TLS certificate*. La seleccionamos para iniciar el proceso.

Seguiremos los pasos indicados para generar un certificado con Let's Encrypt. El sistema también nos ofrece la posibilidad de adquirir un certificado de otras entidades de certificación, pero estas tienen coste.

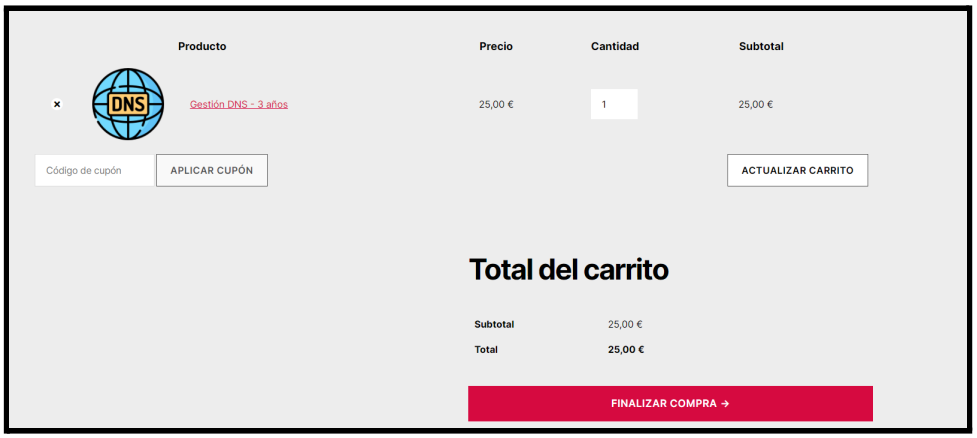
6.4.3. Aplicación del tema. Adición de productos.

Se define como tema del sitio el “Twenty Twenty”, por el estilo moderno de este.

Como página principal, se ha definido la opción “una página estática”, seleccionando una que se ha creado para presentar la empresa. En la página a continuación se puede ver cómo ha quedado. El sitio web está disponible en la dirección https://glez.cloud/.

También se ha desactivado el menú que trae WordPress por defecto, para añadir una serie de páginas arbitrarias:

El carrito de compra es funcional:



Tu hosting de confianza

Mi cuenta Servicios Tienda Contacto Carrito Buscar

GLEZ.CLOUD

Qué ofrecemos

Alojamiento WordPress

Ultra rápido y ultra seguro. Tu sitio web estará siempre online, listo para que tus clientes te encuentren.

Soporte experto y cercano

En GLEZ.CLOUD los Success Strategist son los encargados de ayudar en todo lo que necesiten a los clientes.

Alojamiento de sitios estáticos

Si prefieres subir tu propio código, cuenta con nosotros a tu lado. En GLEZ.CLOUD nos adaptamos a tus necesidades

Qué opinan nuestros clientes

¿Macron? ¿Trudeau? Mi nuevo amor es GLEZ.CLOUD. Me va la página de La Mandíbula que echa humo.

P. Sánchez

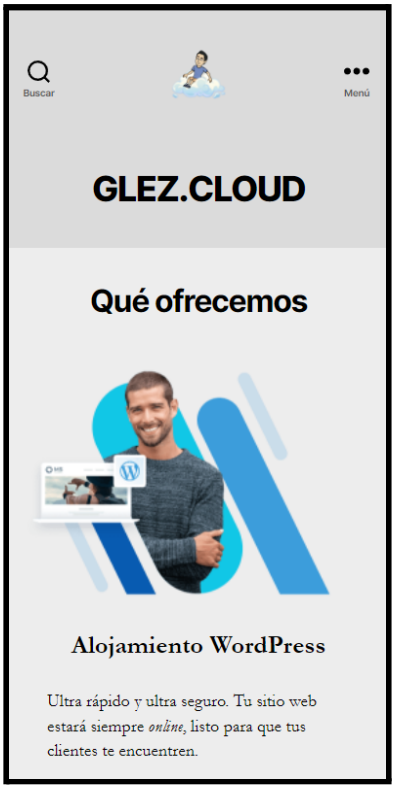
¿Libertad? ¡Libertad es poder usar el CMS que quiera! Con GLEZ.CLOUD puedo.

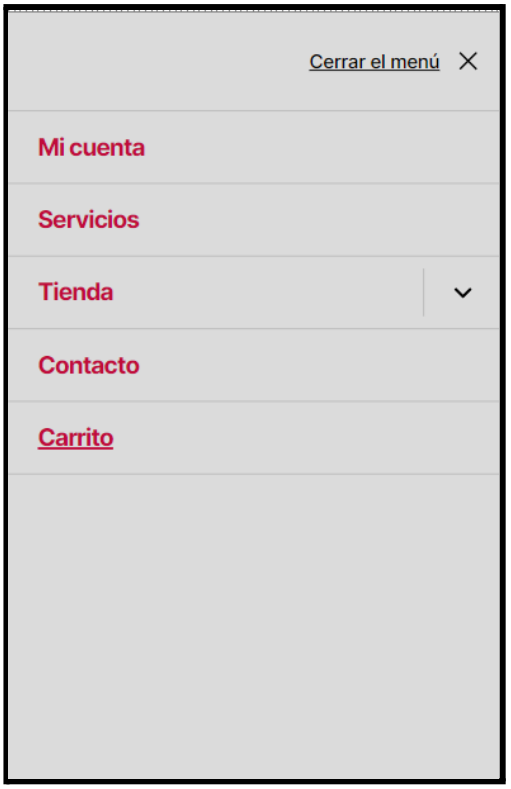
I. Ayuso

¿Independencia? Sí, pero que el equipazo de GLEZ.CLOUD te echa una mano en un puentazo.

G. Rufián

En mobile (emulando un iPhone 12 Pro):





6.4. Creación de planes de hosting y *reselling*

En el momento de la redacción de este documento, el cliente puede adquirir los siguientes productos a través de la página web/tienda de la empresa (https://glez.cloud):

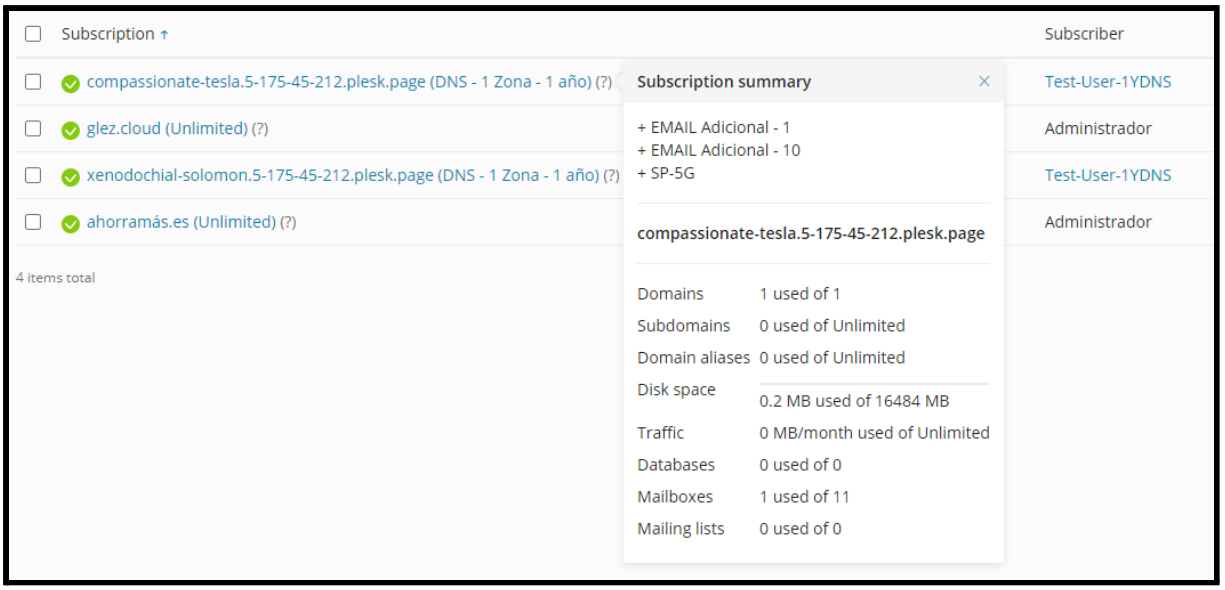
Servicio	Precio normal	Precio rebajado
Gestión DNS - 1 año	9,99 €	N/A
Gestión DNS - 2 años	19,98 €	N/A
Gestión DNS - 3 años	29,97 €	25,00 €
Correo electrónico - 1 cuenta/año	12,00 €	N/A
Correo electrónico - 10 cuenta/año	120,00 €	100,00 €
Alojamiento WordPress - Básico	5,00 €	N/A
Alojamiento WordPress - Avanzado	15,00 €	N/A
Alojamiento WordPress - Sin límites	49,99 €	29,99 €
Alojamiento Drupal - Básico	5,00 €	N/A
Alojamiento Drupal - Avanzado	15,00 €	N/A
Alojamiento Drupal - Sin límites	49,99 €	29,99 €
Alojamiento Joomla - Básico	5,00 €	N/A
Alojamiento Joomla - Avanzado	15,00 €	N/A
Alojamiento Joomla - Sin límites	49,99 €	29,99 €
Alojamiento Estático - Básico	5,00 €	N/A
Alojamiento Estático - Avanzado	15,00 €	N/A
Alojamiento Estático - Sin límites	49,99 €	29,99 €

Los productos disponibles en la página web se corresponden con un plan de servicio en Plesk. Un plan de servicio se une con un dominio para formar una suscripción. Una suscripción pertenece a una cuenta de cliente.

Una suscripción puede tener varios planes de servicio adicionales, llamados add-ons. Esto es de gran utilidad para completar los planes. Supongamos que el cliente contrata únicamente la gestión DNS, pero posteriormente desea también gestionar su correo electrónico y alojar su WordPress con GLEZ.CLOUD. Estas dos

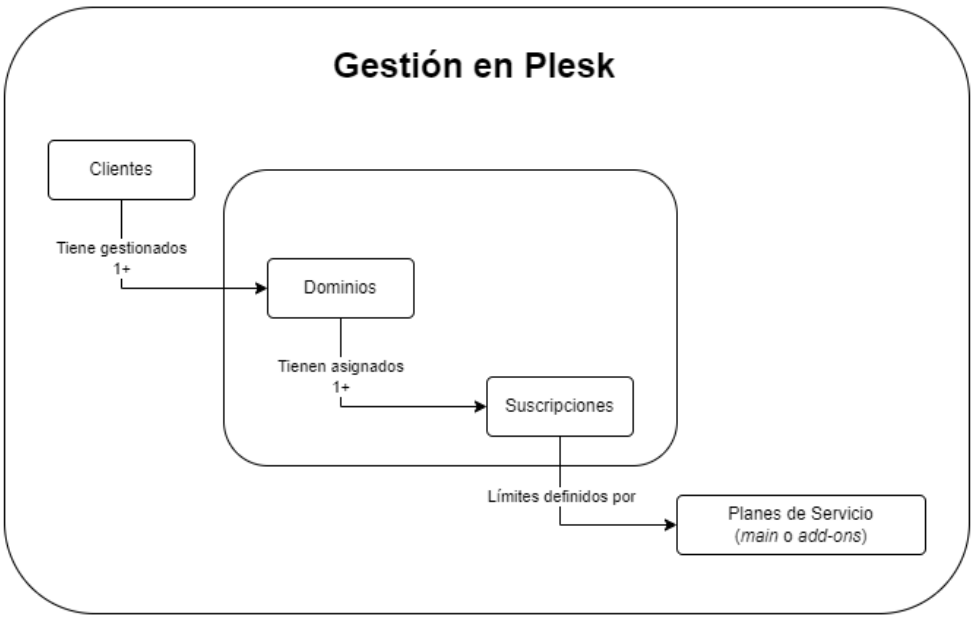
últimas opciones añadidas serían, para Plesk, planes tipo *add-on*, que podríamos traducir como complementos en castellano.

Así es como veríamos una suscripción con add-ons:



Tal y como se puede ver en la imagen anterior, además de su suscripción básica, se han añadido 3 *add-ons*: 2 relativas al servicio de correo electrónico, que añaden en conjunto 11 buzones; y otra para ampliar el espacio de almacenamiento en 5GB.

Para facilitar la comprensión de la relación entre estas entidades, se ha generado el siguiente diagrama:

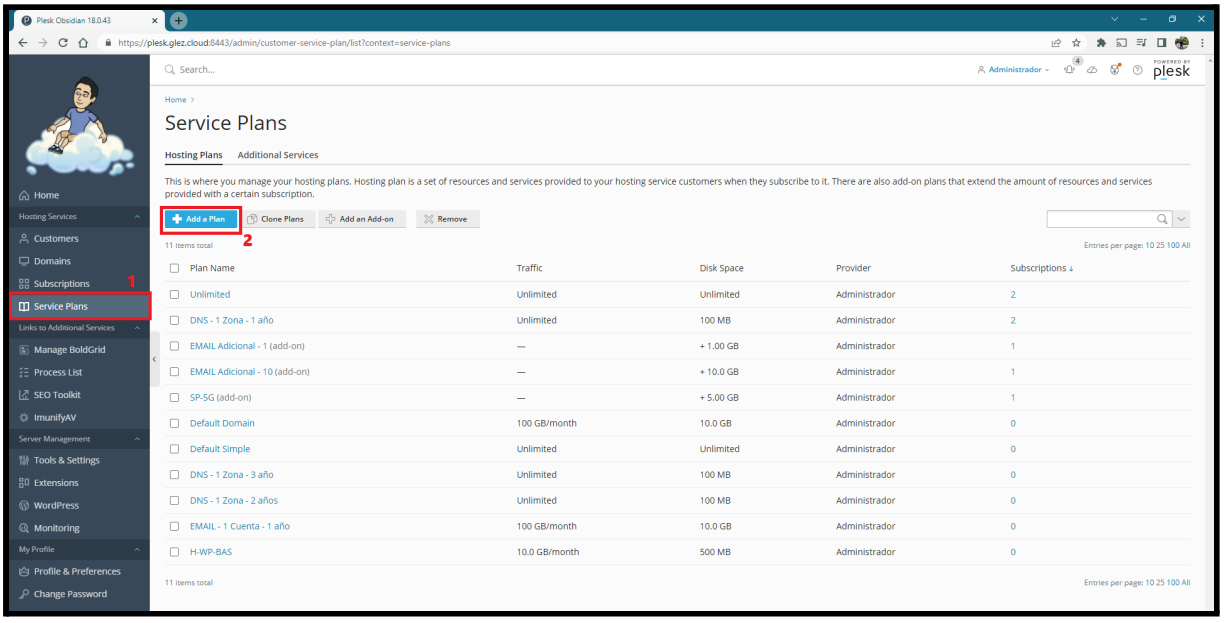


Para ejemplificar la creación de un plan de servicio personalizado generaremos el adecuado para la opción de *Alojamiento WordPress – Avanzado*[109], disponible para ser adquirido en la página web de la empresa. Las características principales son:

Estos son límites:

- Almacenamiento: 2 GB/mes
- Tráfico de red: 50 GB/mes
- Límite de visitas: 50.000 solicitudes/mes

Para gestionar los planes de servicio personalizados, necesitamos acceder al panel con un usuario con permisos de administrador, una vez hecho en el menú lateral seleccionamos la opción *Service Plans*. Para crear uno nuevo, hacemos clic en *Add a Plan* o *Add an Add-on*, seleccionando uno u otro según el tipo de plan que queramos crear en este momento. También tenemos la posibilidad de clonar un plan ya existente.



En este caso, para la opción de *Alojamiento WordPress – Avanzado*, vamos a crear un nuevo Plan. El nombre será *H-WP-AVAN*. Este nombre tiene importancia, pues según vayamos creando planes en la plataforma, será más difícil administrarlos si

[109] https://glez.cloud/producto/alojamiento-wordpress-avanzado/

no hemos definido correctamente la nomenclatura. La nomenclatura de los planes se puede resumir en la siguiente tabla:

Tipo de servicio	Subtipo de servicio	Tamaño/Duración
H (Hosting)	WP (WordPress)	BAS (Básico)
H	WP	AVAN (Avanzado)
H	WP	UNLIM (Ilimitado)
H	DR (Drupal)	BAS/AVAN/UNLIM
H	JO (Joomla)	BAS/AVAN/UNLIM
H	ST (Estático)	BAS/AVAN/UNLIM
SP (Espacio en disco)	(No aplicable)	5G (5 GB de espacio adicional)
BW (Transferencia)	(No aplicable)	10G (10 GB de espacio adicional)
EMAIL	1 Cuenta	1 Año
EMAIL Adicional	(No aplicable)	1
EMAIL Adicional	(No aplicable)	10
DNS	1 Zona	1 año
DNS	1 Zona	2 año
DNS	1 Zona	3 año

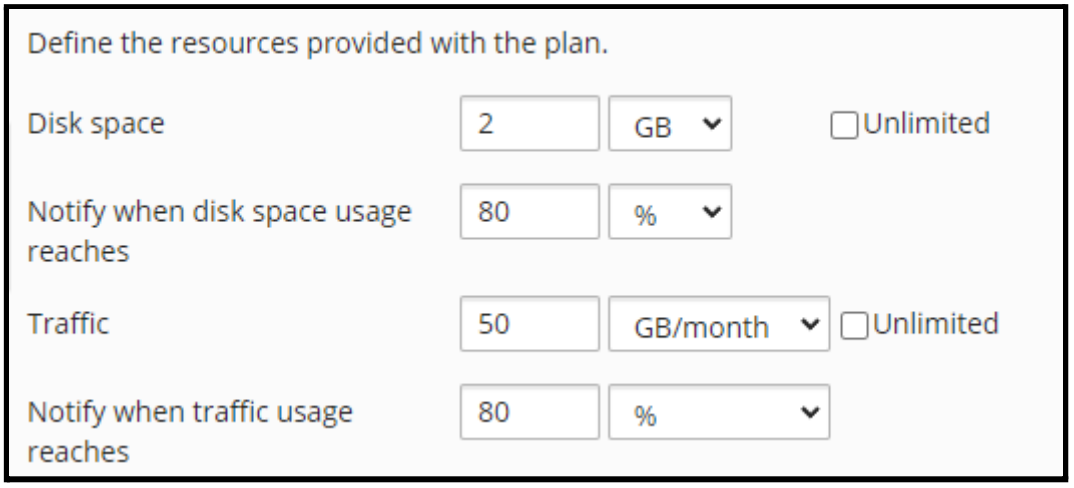
Continuando con la creación del plan de servicio personalizado para *H-WP-AVAN*, lo primero que debemos introducir es el nombre en el campo *Service plan name*.

En la pestaña ***Resources***, debemos añadir/modificar los siguientes valores:

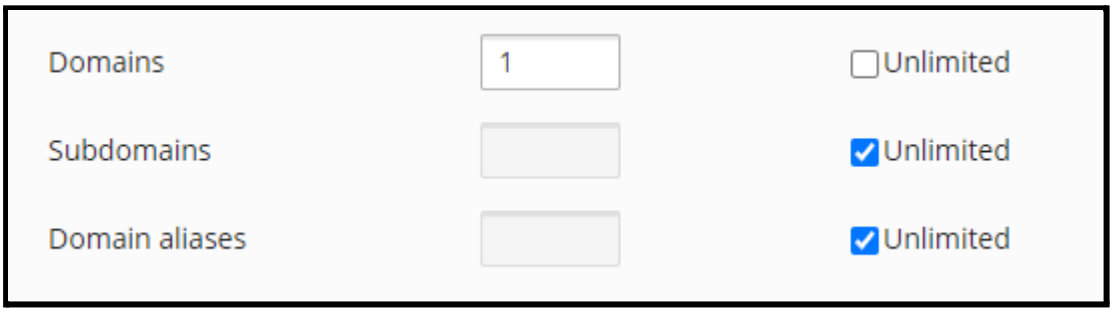
- En tanto a la política de *overuse*, que define cómo se actuará cuando se saturen los límites del plan, hemos definido que sí se permita al usuario sobrepasar el espacio en disco y tráfico. El motivo es simple: en GLEZ.CLOUD queremos ver crecer a nuestros clientes y estaremos

encantados de que una campaña de marketing tenga tanto efecto que el plan inicialmente contratado se quede corto. En cualquier caso, el equipo técnico estará en contacto permanente con el cliente y su *success strategist*.

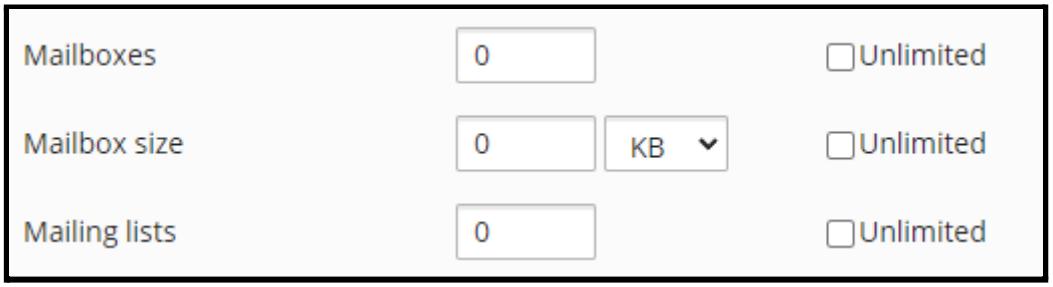
- Para los límites de espacio en disco y tráfico definimos los indicados en la ficha del producto. Se activa la opción de enviar una notificación cuando el uso llegue al 80%



- El plan sólo permite añadir un dominio principal. Aunque se podrán añadir subdominios y alias de dominio de forma ilimitada.



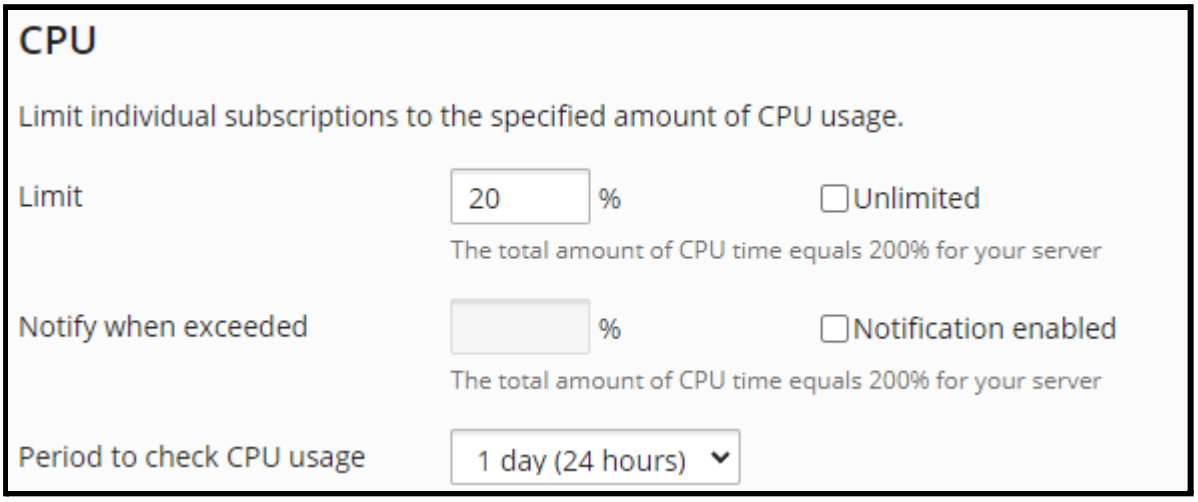
- No incluye cuentas de correo electrónico, aunque por supuesto el cliente puede contratar el add-on para disponer de este servicio.



- Se incluye 1 sitio web de WordPress y toda la gama de servicios relacionados con este producto, como Backups y *Smart Update* para plugins.

En la pestaña ***RAM, CPU, Disk I/O***, debemos añadir/modificar los siguientes valores:

- Para el uso de CPU, se ha fijado en un 20%. Es de un 10% para los hosting básicos y de un 30% para los hosting ilimitados. Puesto que el servidor tiene 2 vCPUs, el total no es 100% sino 200%. El uso de CPU es comprobado cada 24 horas.



- En tanto a la RAM, el límite para este nivel se ha fijado en 300 MB. 200 MB para básico y 400 MB para ilimitado.

En la pestaña ***Permissions***, debemos añadir/modificar los siguientes valores:

- Por un lado, no permitimos que el usuario pueda gestionar el servidor mediante SSH, por seguridad.
- Sí permitimos la gestión DNS, pues viene incluida con todos los planes de servicio. También activamos la gestión de los ajustes del hosting (SSL/TLS, páginas de error personalizadas, etc) y de PHP. El resto de las opciones las mantenemos por defecto.

En la pestaña ***Hosting parameters***, debemos añadir/modificar los siguientes valores:

- Definimos la “cuota dura” de almacenamiento en algo más de lo contratado, para que el corte no sea por sorpresa.

- Permitimos el uso de SSL/TLS y de la redirección 301 segura para SEO desde HTTP a HTTPS.
- Activamos las estadísticas de tráfico y las páginas de error personalizadas.
- En tanto a la base de datos predeterminada, elegimos el servidor MariaDB del propio Plesk.

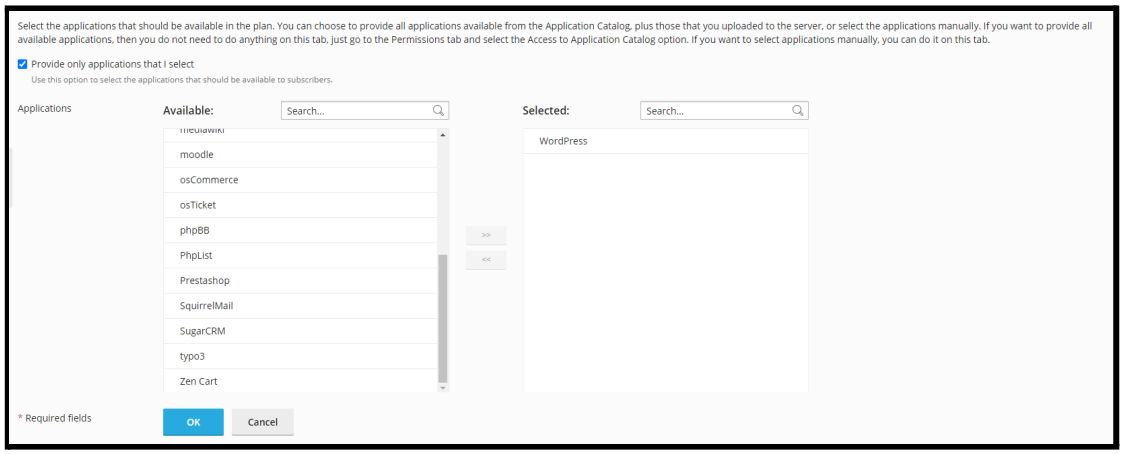
En la pestaña ***PHP Settings***, debemos añadir/modificar los siguientes valores:

- Por seguridad, la versión de PHP con la que funcionará el hosting será la 8.0.18, pudiendo elegir una anterior según las necesidades del cliente.
- Los límites de memoria y ejecución se mantienen. Únicamente se amplía el tamaño máximo de archivo a 16 MB (por defecto a 2 MB), para evitar que los usuarios se topen con un mensaje de error al intentar subir a su sitio web imágenes o documentos pesados.

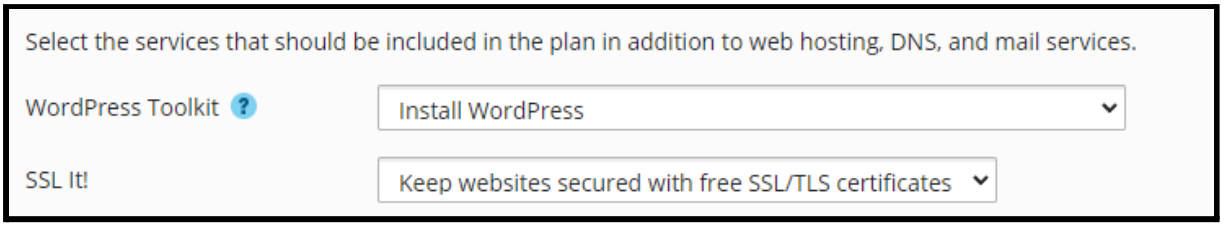
En la pestaña ***Web Server***, no modificamos nada. En la pestaña ***Mail***, desactivamos el servicio de correo electrónico, pues no está incluido en este plan de servicio. El cliente siempre puede solicitar su activación mediante su Success Strategist o realizando la compra en la tienda.

En la pestaña ***DNS***, se define el servidor como maestro, aunque el cliente puede no utilizar nuestro servicio como servidor DNS principal. En la pestaña ***Logs & Statistics***, se define que las estadísticas de tráfico se mantengan durante 3 meses y los archivos de registro sean rotados cada 10 MB, comprimiendo los antiguos.

En la pestaña ***Applications***, únicamente se permite al usuario la instalación de Wordpress al ser este el único servicio contratado.



La pestaña ***Additional Services***, que quizá puede pasar desapercibida es muy importante. Aquí seleccionamos que se instale automáticamente WordPress y además se securice el sitio con un certificado SSL/TLS.



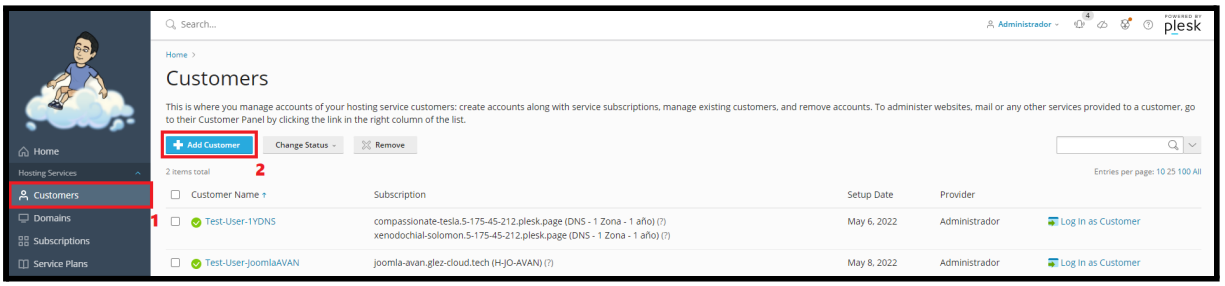
6.5. Creación de clientes de ejemplo

Para esta sección se crearán varios clientes, con una o varias suscripciones. Las credenciales de todos los clientes, tanto en Plesk como en los propios sitios instalados están disponibles en el Anexo III de este documento como referencia.

Para ejemplificar la facilidad en la gestión, ejemplificaremos el proceso para un **H-WP-AVAN** y para un **H-DR-BAS.**

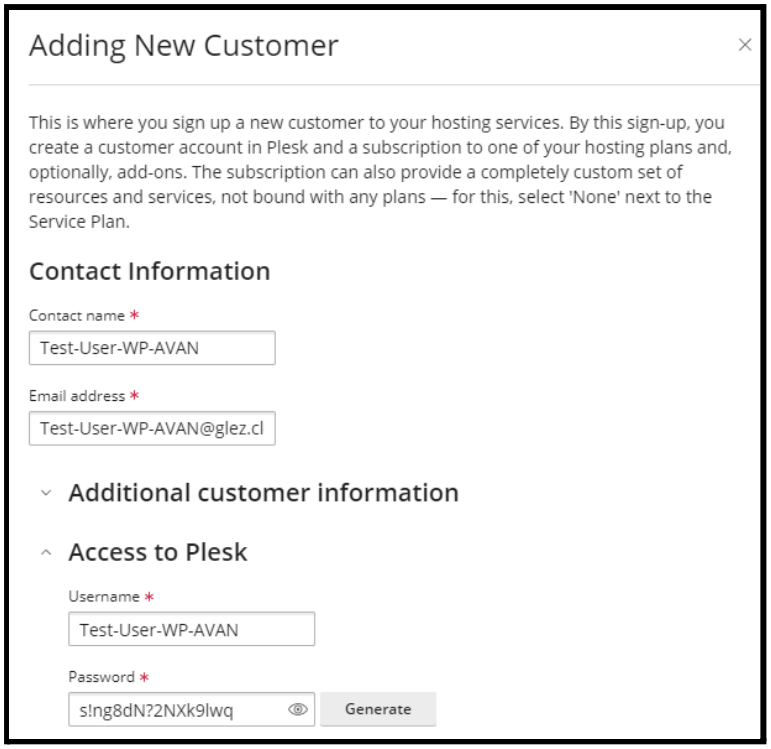
6.5.1. Suscripción de H-WP-AVAN

Desde la sección *Customers*, donde podemos ver los clientes existentes, hacemos clic en *Add customer* para proceder a agregar uno nuevo.



Al hacer clic en este botón, se nos desplegará desde el lateral izquierdo el asistente. Indicaremos el nombre de contacto así como la dirección de correo electrónico del usuario a crear (para simplificar la operativa, se ha tomado la decisión de que todos estén bajo el subdominio glez.cloud).

Para el acceso al panel web de Plesk para la gestión y administración del dominio y los servicios asociados, generamos una contraseña, tal y como se puede ver en la imagen a la izquierda de este texto.

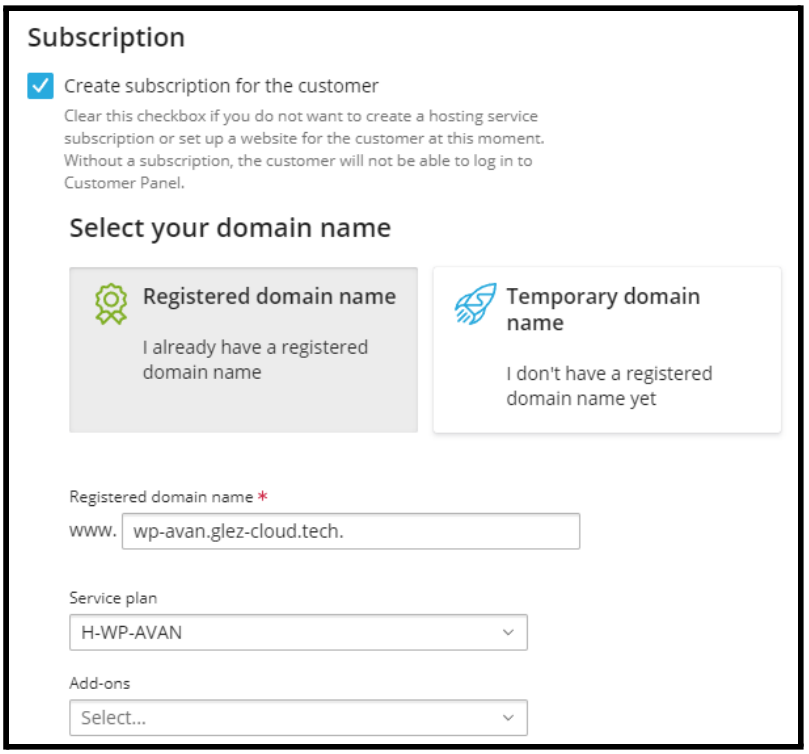


A continuación, haciendo *scroll* hacia abajo en el asistente veremos las opciones de suscripción.

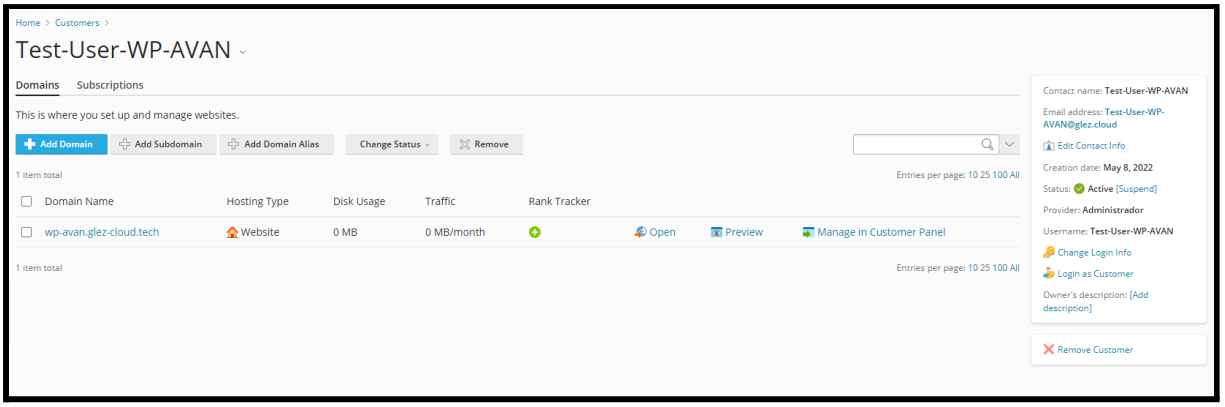
La primera decisión que debemos tomar es respecto al nombre de dominio: indicar uno propio o usar uno que Plesk nos provea dentro del TLD *plesk.page*. Puesto que tenemos el dominio gles-cloud.tech contratado para estos usos, lo usaremos.

Indicamos por tanto un subdominio de este TLD (la/las entradas/s DNS deben ser añadidas en el servidor DNS, OVH).

En tanto al *Service Plan,* indicaremos por supuesto *H-WP-AVAN* puesto que estamos ante un Hosting tipo WordPress en versión Avanzado. No vamos a seleccionar add-ons en este momento, siempre podemos hacerlo más adelante.



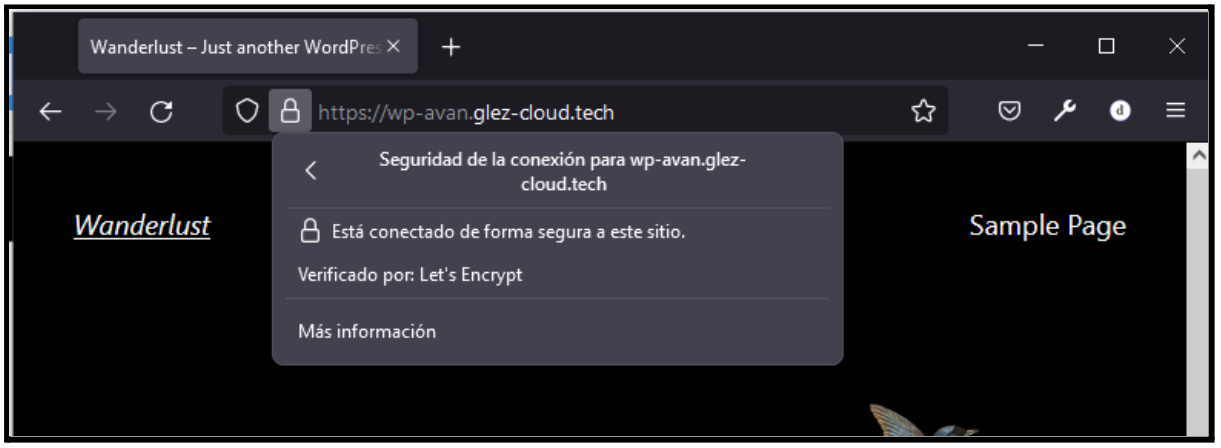
Hacemos clic en *Add customer* para finalizar el proceso. Una vez terminada la instalación, que durará unos pocos segundos, veremos el cliente y su dominio en nuestro panel:



Puesto que hemos seleccionado la opción (a nivel de Service Plan) de instalar automáticamente WordPress e instalar un certificado SSL/TLS, no debemos preocuparnos de nada más.

Para ver lo mismo que vería el usuario podemos usar uno de los siguientes botones en la sección de Dominio o Usuarios, respectivamente:

Podemos comprobar, accediendo a wp-avan.glez-cloud.tech como efectivamente, el CMS WordPress y el certificado han sido desplegados de forma satisfactoria:



Desde la parte superior de la pantalla, estando la sesión iniciada como administrador vemos una serie de opciones. Vamos a describirlas en este momento:

Con la primera opción, marcada con “1” en la imagen, retornamos a la vista global de Administración, saliendo de la simulación del panel de cliente.

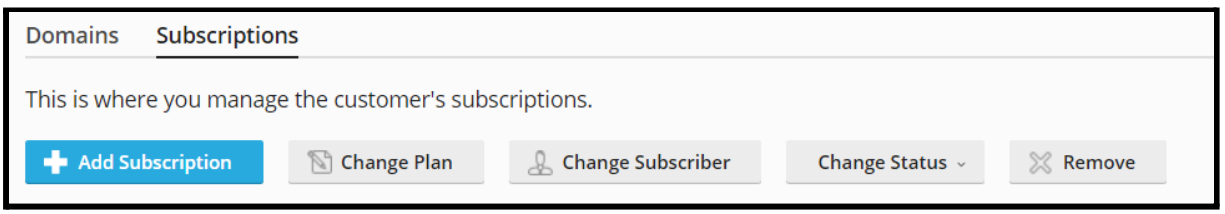
Con la segunda opción, marcada con “2” en la imagen, podemos cambiar entre usuarios del sistema Plesk.

Con la tercera opción, marcada con “3” en la imagen, podemos cambiar entre los distintos sitios que el usuario actual (*Test-User-WP-AVAN*, en este caso) tenga dados de alta en el sistema Plesk.

6.5.2. Suscripción de H-DR-BAS

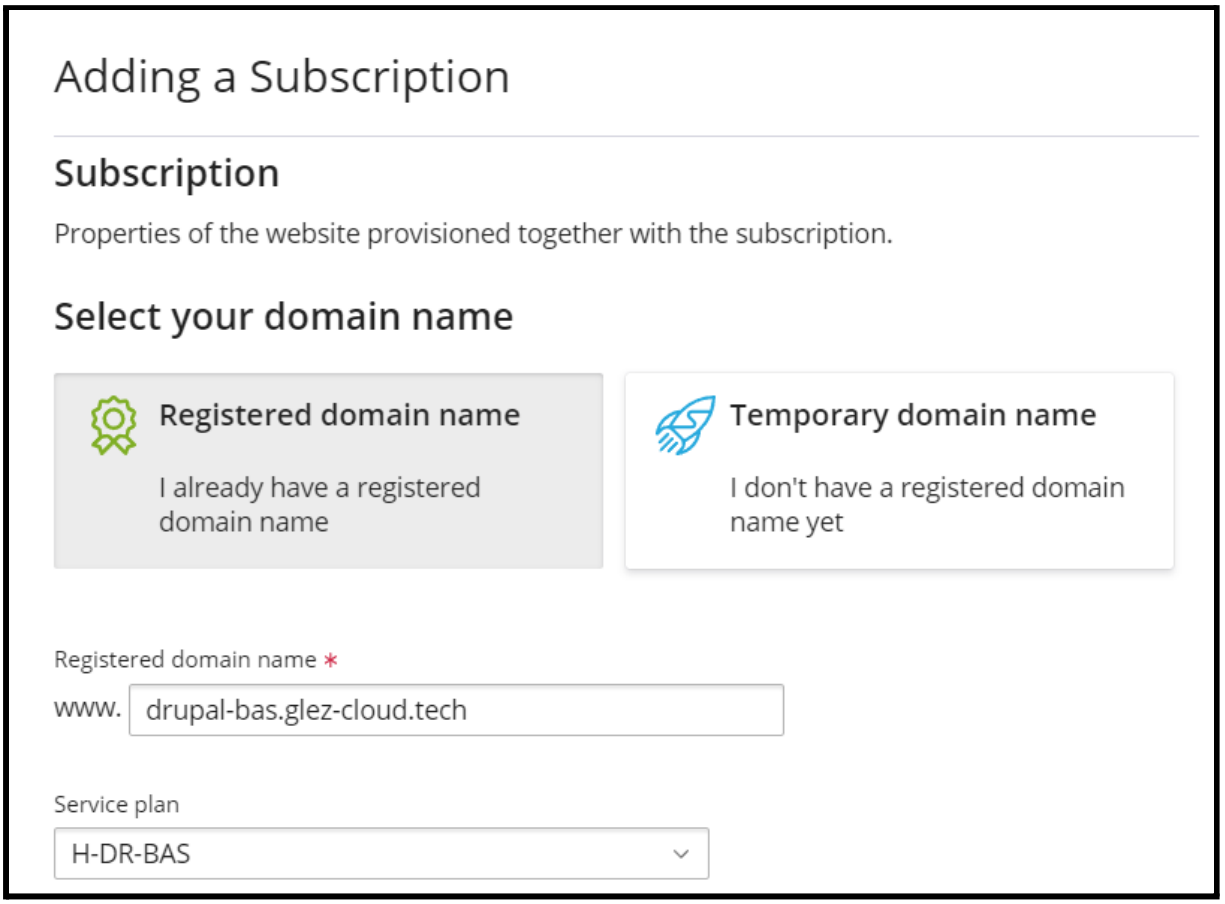
Para esta nueva suscripción no vamos a añadir un nuevo usuario, sino que la añadiremos al usuario de ejemplo usado para la sección anterior (Test-User-WP-AVAN).

Para hacerlo nos dirigimos en el Panel de Administración hasta *Customer*. Entre la lista de todos los usuarios posibles, seleccionamos al que deseamos añadir la suscripción. En la parte superior, debajo del nombre del cliente, podemos ver dos “pestañas”: *Domains* y *Subscriptions*. Hacemos clic sobre la segunda opción. Para añadir una nueva usamos el botón *Add Subscription*:



Cuando hacemos clic en este, se nos abre un asistente similar al visto en la sección anterior. Nos ofrecerá varias opciones: *Blank website*, *Upload files*, *Deploy using Git*, *WordPress Site* e *Import website*. Seleccionamos *Blank website*, pues luego "machacaremos" todo con la instalación de Drupal.

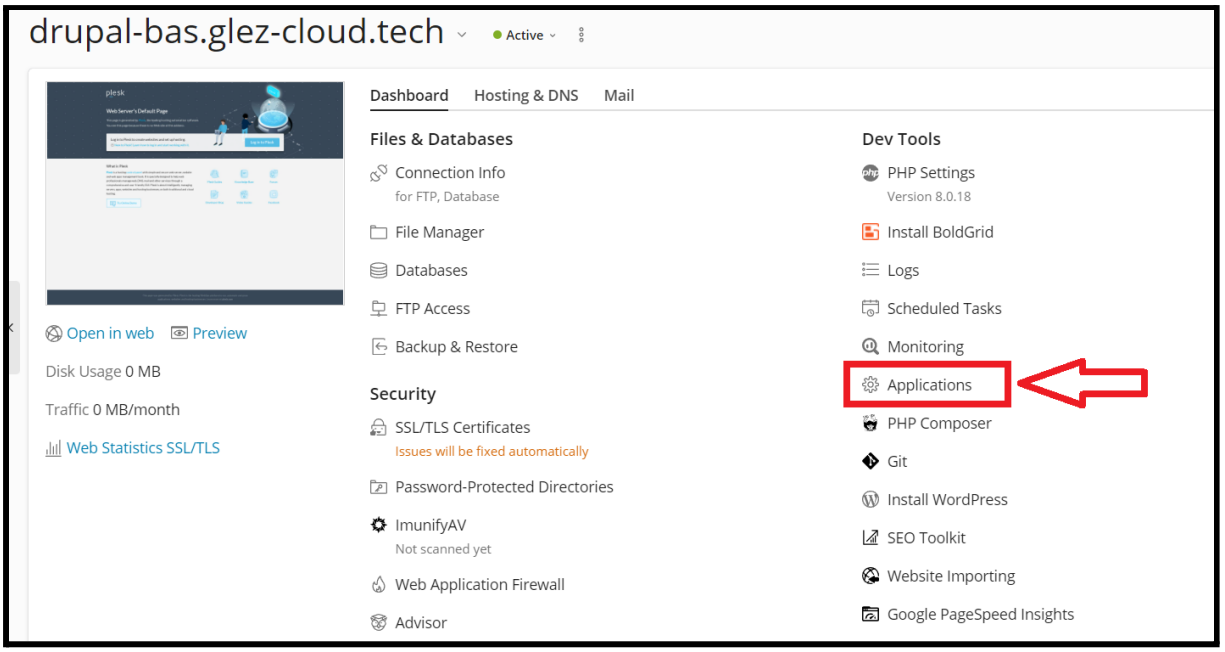
De forma similar a las veces anteriores, debemos indicar el dominio y el *Service Plan* para esta suscripción:



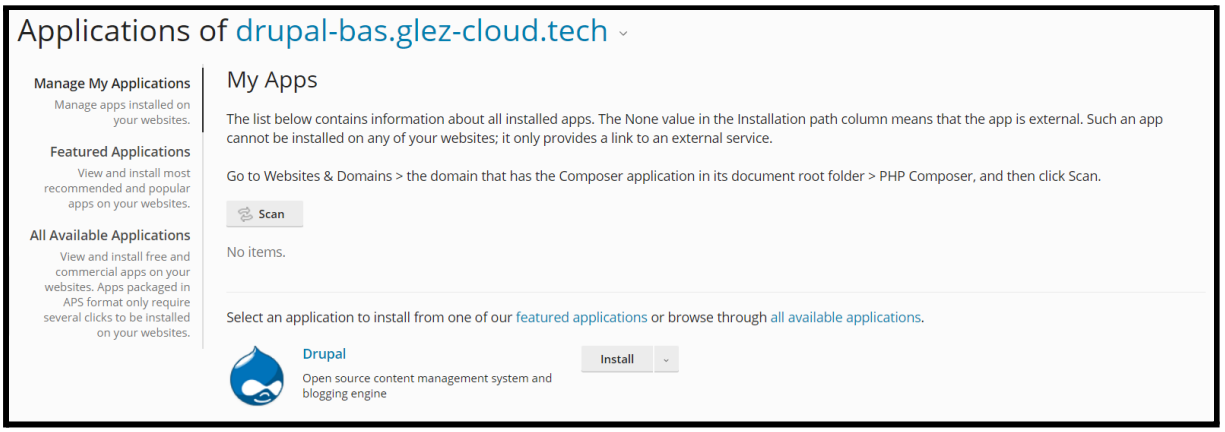
Sin embargo, a partir de aquí el proceso difiere ligeramente, pues el sistema no instala de forma automática Drupal en el sitio del cliente. Esto no quiere decir que debamos desplegar la aplicación de forma manual, sino que hay que indicar a Plesk que la instale desde el menú de gestión web.

Para hacerlo, desde la pestaña de gestión del dominio, seleccionamos "*Applications*". Esta opción la encontraremos, tanto desde nuestro panel de Administración como emulando el panel del cliente. De hecho, el propio cliente

también podría instalar Drupal, pues lo hemos permitido (la única aplicación permitida para ser instalada) en la configuración del *Service Plan*.



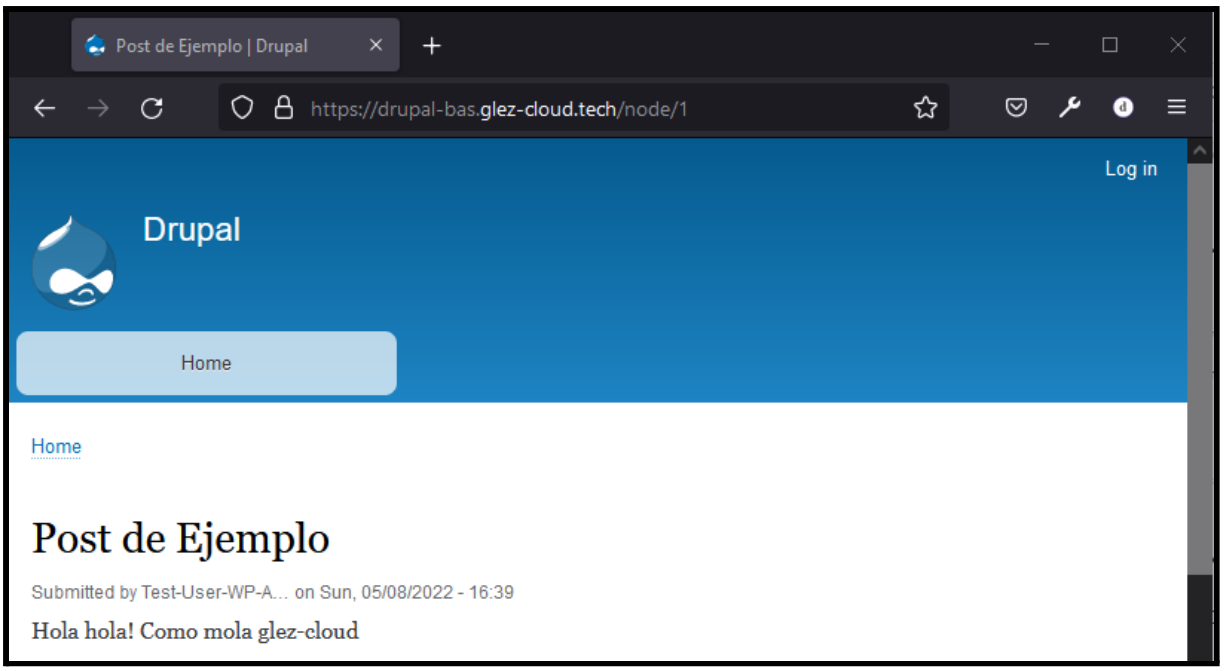
Como se ha indicado anteriormente, sólo puede ser instalado Drupal[110]. Hacemos clic en *Install* para iniciar el proceso:



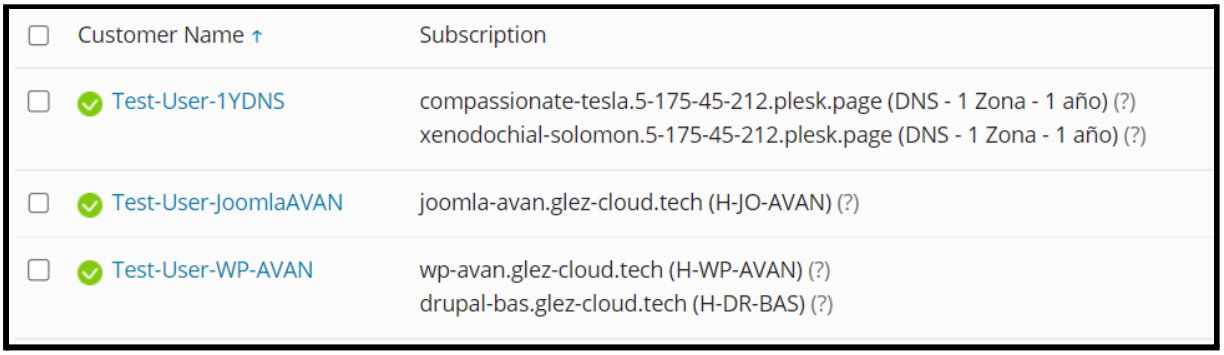
Al terminar el proceso de instalación, que tardará unos pocos segundos, se nos mostrará el usuario y la contraseña generada.

Se ha comprobado como estas eran válidas, añadiendo un artículo al CMS. También se ha comprobado la correcta instalación del certificado SSL/TLS:

[110] Lamentablemente, Drupal no es compatible con la última versión de PHP. En las suscripciones de este CMS se ha seleccionado la versión 7.4.29 en lugar de 8.0.18.



Si en este momento nos desplazamos a la pantalla de gestión de clientes, veremos la siguiente información:



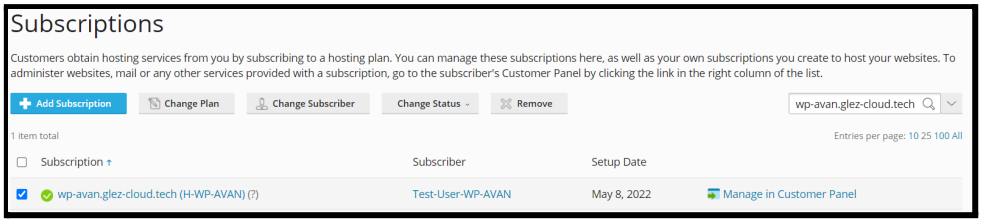
Como se puede observar, el cliente *Test-User-WP-AVAN* dispone de 2 suscripciones: H-WP-AVAN y H-DR-BAS.

6.5.3. Adición de add-on a una suscripción

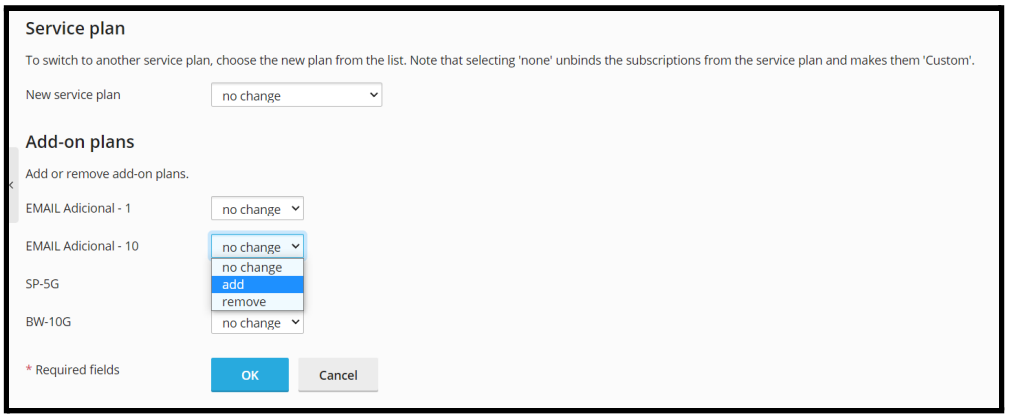
Supongamos en este momento que el cliente Test-User-WP-AVAN decide que desea contratar, además del alojamiento WordPress, 10 cuentas de correo electrónico para sus usuarios bajo el dominio `wp-avan.glez-cloud.tech`.

Para completar esta petición, debemos añadir una suscripción de tipo *add-on* a la suscripción principal. En la sección *Subscriptions* usamos el buscador situado en la

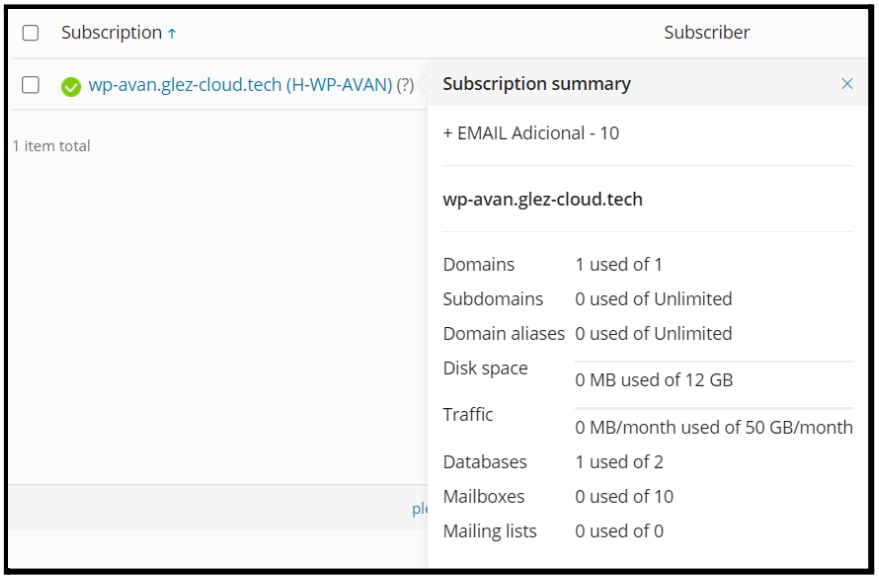
parte superior derecha de la pantalla para localizar la suscripción a la que queremos añadir el *add-on*. Una vez localizada, usamos la casilla a la izquierda del dominio para seleccionarlo. En la parte superior de la pantalla, veremos la opción de *Change Plan*. Hacemos clic sobre esta.



Al hacerlo nos aparecerá la siguiente pantalla:



Desde esta, hacemos clic en el *add-on* deseado para seleccionar la opción *add*. Hacemos clic en *OK* para guardar los cambios. El resumen muestra los cambios aplicados:



Al acceder a su panel de gestión en Plesk, el cliente ya puede ver en el menú lateral izquierdo la opción “Correo”. Desde aquí, puede gestionar todo lo relacionado con el servicio de correo electrónico.

Puesto que la instalación del servicio se ha hecho con posterioridad al momento de creación del dominio en el sistema, el servicio se encuentra deshabilitado.

Basta con marcarlo (usando la casilla a la derecha del dominio) y después usar el botón *Activar/desactivar servicios*.

Al hacerlo, nos aparecerá en la pantalla una ventana en la debemos indicar que efectivamente, deseamos activar el servicio de correo electrónico.

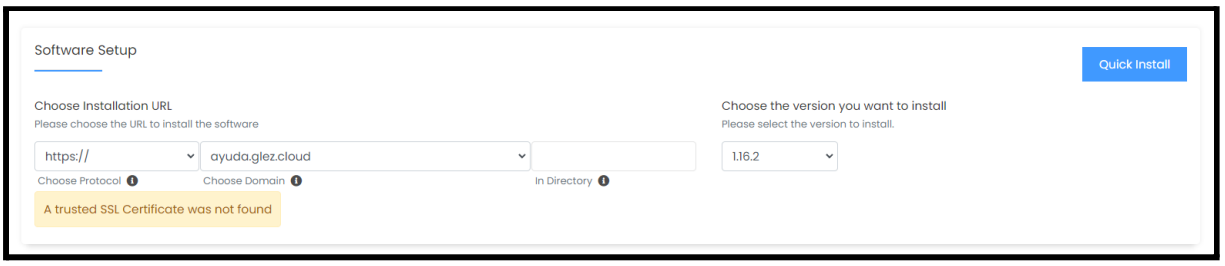
También nos ofrece la posibilidad de activar la protección DKIM, sobre la protección DKIM, puede ser útil la consulta del Anexo IX: Seguridad en el correo electrónico: DKIM, SPF y DMARC de este documento.

6.6. osTicket como plataforma de soporte

6.6.1. Instalación de osTicket

osTicket será la plataforma de soporte, mediante la cual se atenderán las consultas técnicas de los clientes, así como la plataforma para dar seguimiento a las solicitudes comerciales y pedidos a través de la web.

La instalación se realizará usando softaculous, que simplifica en gran medida la instalación de software. Tal y como se puede ver en la imagen a continuación, será accesible desde la dirección https://ayuda.glez.cloud

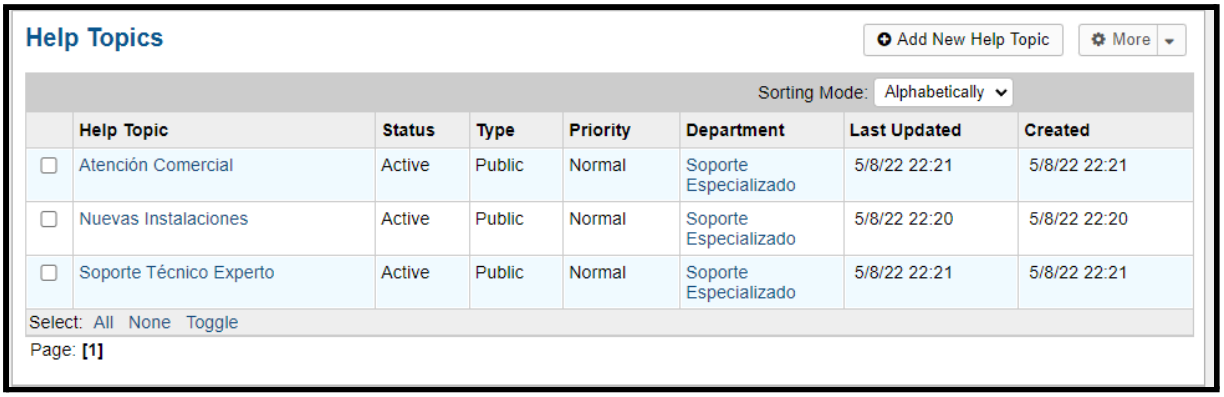


Una vez instalado el aplicativo, queda pendiente configurarlo para ajustarlo a nuestras necesidades.

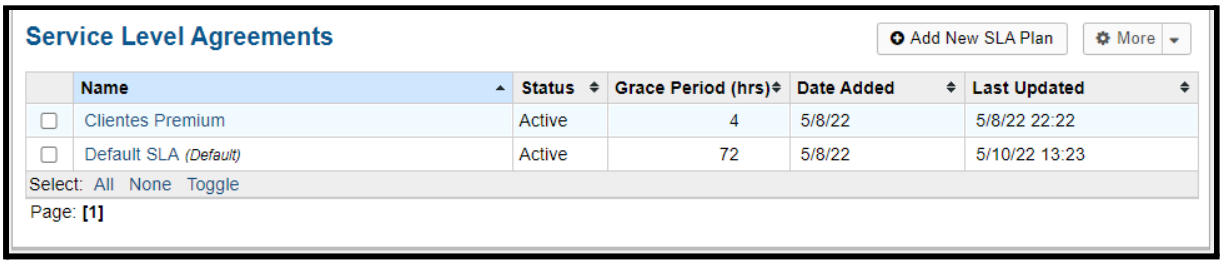
6.6.2. Configuración y personalización de osTicket

- Se cambia el nombre del centro de ayuda a “GLEZ.CLOUD - Ayuda” para que sea más amigable y concuerde con el nombre de la empresa.
- Se activa el acceso siempre por HTTPS (habiendo generado de forma previa el certificado SSL/TLS usando la integración de Plesk con Let’s Encrypt).
- Se amplía el límite de los archivos adjuntos hasta los 2 MB, que es suficiente para la mayoría de capturas de pantalla y archivos.
- Se modifica el logo de osTicket para incluir el de GLEZ.CLOUD.

- Los identificadores de los tickets se mantienen aleatorios, aunque reduciendo de 6 a 4 los dígitos que se usarán para su generación.
- Se desactiva el límite de tickets abiertos por usuario y se activan las notificaciones de confirmación a los clientes que abran nuevos tickets.
- Se añaden *Help Topics*, "categorías", para poder organizar los tickets en base a estas categorías:



- Se agrega el SLA *Clientes premium* con una duración de 4 horas y se modifica el SLA por defecto a 72 horas:

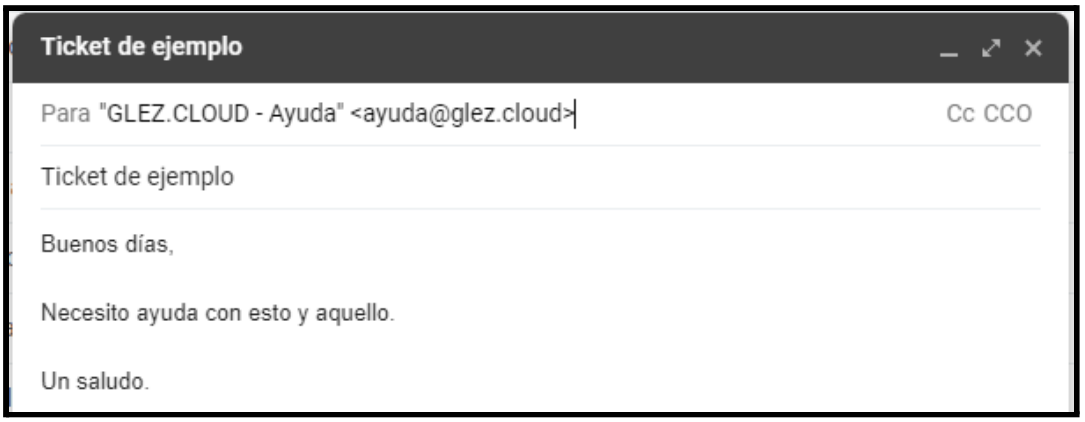


- Se modifican las páginas públicas, traduciéndolas al castellano y añadiendo textos aplicables a GLEZ.CLOUD. Los cambios son visibles, por ejemplo, en la página de inicio del sistema de soporte:

- Se configura el envío de correo electrónico a través del SMTP de Google Workspace, puesto que el dominio glez.cloud ya está dado de alta en el servicio. Se ha comprobado como los mensajes llegan correctamente, cumpliendo todas las directivas de seguridad.

- También se configura el sistema para permitir la generación de nuevos tickets con el envío de un correo electrónico a ayuda[@]glez.cloud.

  Este mensaje de correo electrónico se convierte en un ticket:



  El ticket creado lo podemos ver en unos instantes en nuestra web de soporte:

  El cliente recibe la siguiente confirmación de que el ticket ha sido creado correctamente en la plataforma de soporte:

- Se traducen las plantillas de los mensajes de correo electrónico al castellano.

6.7. User Experience (UX)

Una vez el/la cliente ha completado el pedido a través de la página web, se inicia todo el proceso. El proceso de compra a través de WooCommerce/WordPress es realmente sencillo, una compra más a las que tanto estamos acostumbrados/as en estos días.

Al recibirse el pedido, el sistema envía un correo electrónico (a través de Sendinblue) tanto al cliente como a la dirección de correo definida como administrador de la tienda. Ambos correos electrónicos son personalizables.

En el caso del correo electrónico enviado al cliente, que es realmente similar al enviado al administrador, contiene:

- Un mensaje de agradecimiento por la compra realizada
- Los datos para realizar el ingreso bancario
- Un resumen del pedido con los productos adquiridos y el precio unitario de cada uno de ellos.

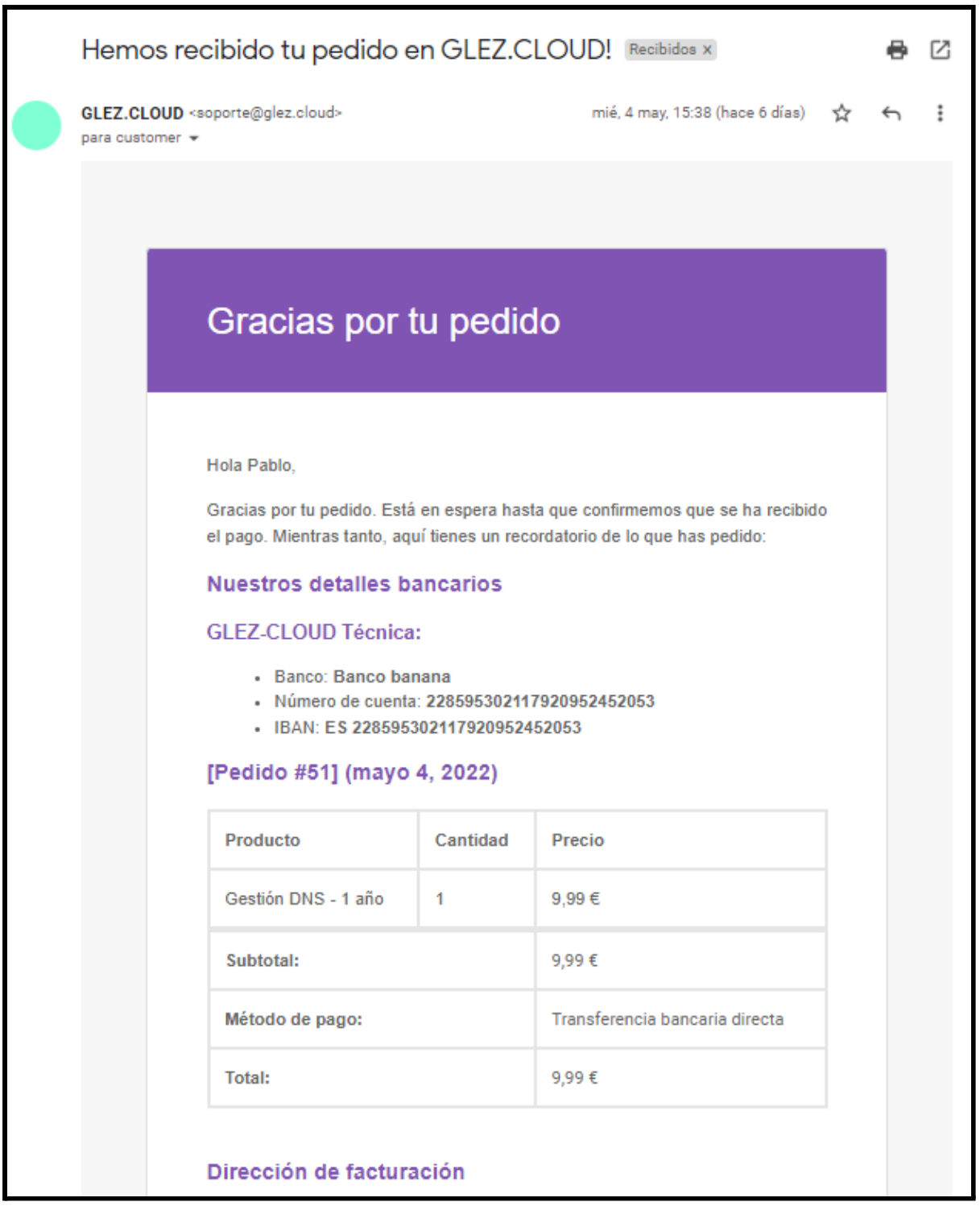
El pago se solicita por transferencia bancaria puesto que al ser una tienda de demostración no se desea poder cobrar mediante tarjeta de crédito. La configuración es realmente sencilla, ya sea con Redsys, Stripe o Woocommerce Payments.

Si bien Redsys es algo más costoso (en tiempo y dinero) al depender de autorizaciones de banco, Stripe y Woocomerce son servicios de pago reconocidos internacionalmente y realmente fáciles de desplegar.

Una vez el agente de GLEZ.CLOUD ha comprobado que la transferencia se ha hecho efectiva, cambia el pedido a en curso.

En paralelo, se crea un ticket en la plataforma de soporte osTicket para realizar desde ahí un seguimiento contínuo.

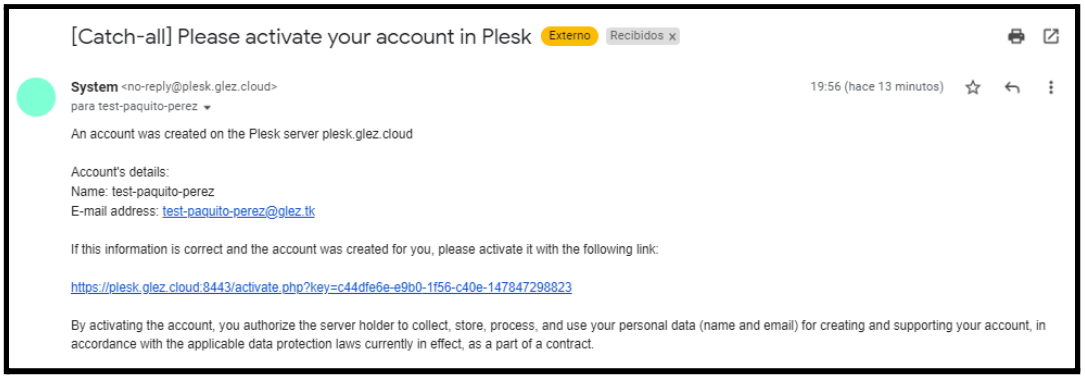
A continuación podemos ver un ejemplo de correo electrónico recibido por cliente al momento de realizar un pedido a través de la tienda online alojada en glez.cloud:



Las cabeceras de este correo se encuentran disponibles en este enlace[111] para su descarga desde el repositorio de GitHub. El correo que recibe el administrador también puede ser descargado desde este enlace[112].

[111] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/4-plesk/correo-compra-customer.eml.txt
[112] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/4-plesk/correo-compra-admin.eml.txt

El primer paso es dar de alta el usuario en Plesk (siguiendo los pasos descritos anteriormente, en la sección 6.4 de este documento) este recibe el siguiente correo electrónico para confirmar su cuenta:


[Catch-all] Please activate your account in Plesk Externo Recibidos x

System <no-reply@plesk.glez.cloud> 19:56 (hace 13 minutos)
para test-paquito-perez

An account was created on the Plesk server plesk.glez.cloud

Account's details:
Name: test-paquito-perez
E-mail address: test-paquito-perez@glez.tk

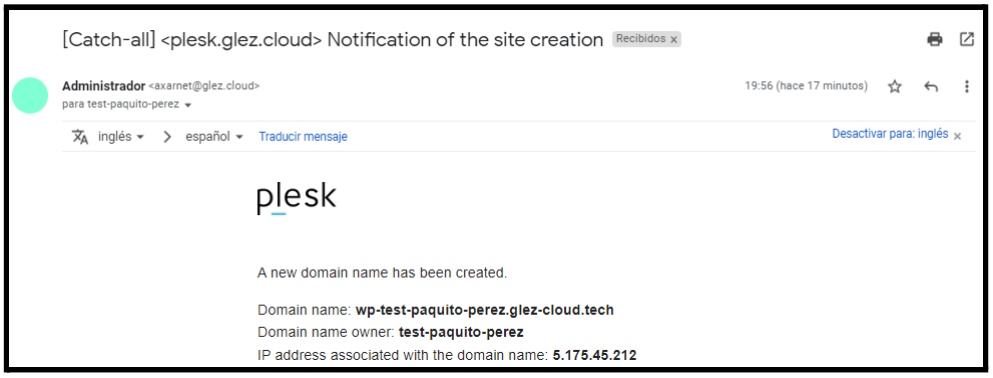
If this information is correct and the account was created for you, please activate it with the following link:

https://plesk.glez.cloud:8443/activate.php?key=c44dfe6e-e9b0-1f56-c40e-147847298823

By activating the account, you authorize the server holder to collect, store, process, and use your personal data (name and email) for creating and supporting your account, in accordance with the applicable data protection laws currently in effect, as a part of a contract.

La recepción de este mensaje de correo electrónico es realmente opcional y depende de lo que seleccione el agente en el *backoffice*. Como norma general no se enviarán esta clase de correo, al utilizarse osTicket para la gestión desde inicio a fin de la relación con los clientes, incluido el proceso de alta y migración.

En la creación de cada uno de los sitios se recibe el siguiente correo electrónico:


[Catch-all] <plesk.glez.cloud> Notification of the site creation Recibidos x

Administrador <axarnet@glez.cloud> 19:56 (hace 17 minutos)
para test-paquito-perez

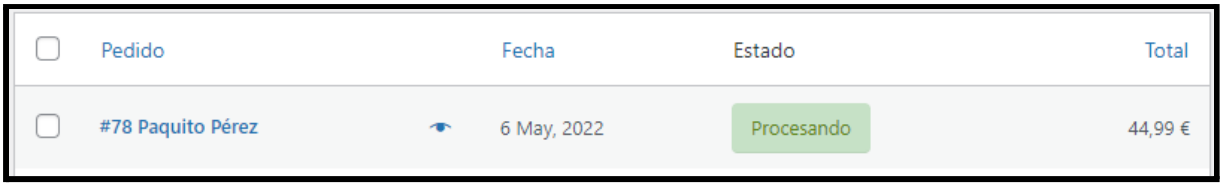
inglés > español Traducir mensaje Desactivar para: inglés x

plesk

A new domain name has been created.

Domain name: **wp-test-paquito-perez.glez-cloud.tech**
Domain name owner: **test-paquito-perez**
IP address associated with the domain name: **5.175.45.212**

Se usará el siguiente pedido para ejemplificar el proceso. Como se puede ver, puesto que el estado es “Procesando”, el pago ha sido confirmado. El agente que ha confirmado el pedido es el encargado de crear el ticket en la plataforma de soporte. Si este no pudiera darle seguimiento (por no poseer los conocimientos técnicos para hacerlo), asignará el ticket al grupo de soporte que sí pueda hacerlo.



Pedido	Fecha	Estado	Total
#78 Paquito Pérez	6 May, 2022	Procesando	44,99 €

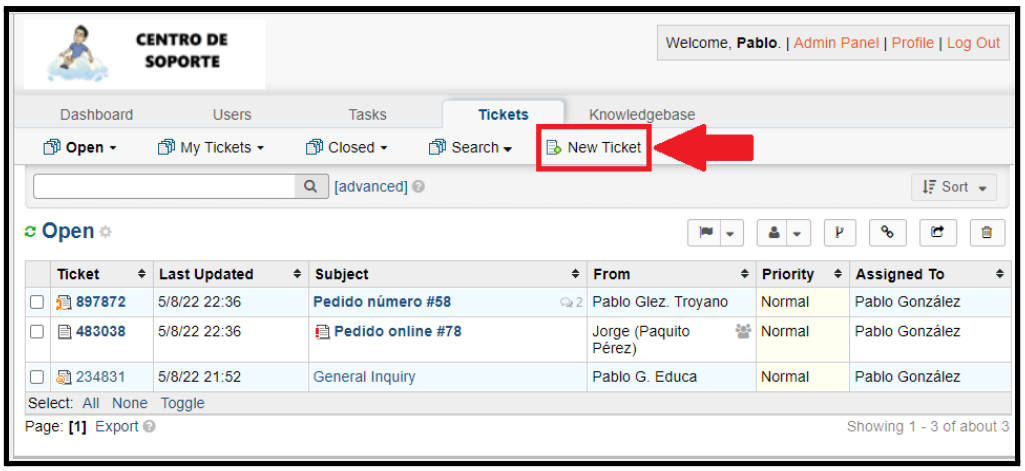
Entrando a los detalles del mismo (basta con hacer clic sobre el número de pedido desde el *backoffice* de Woocommerce) vemos los detalles de este:

Artículo	Coste	Cantidad	Total
Alojamiento Drupal - Sin límites	29,99 €	× 1	29,99 €
Alojamiento WordPress - Avanzado	15,00 €	× 1	15,00 €

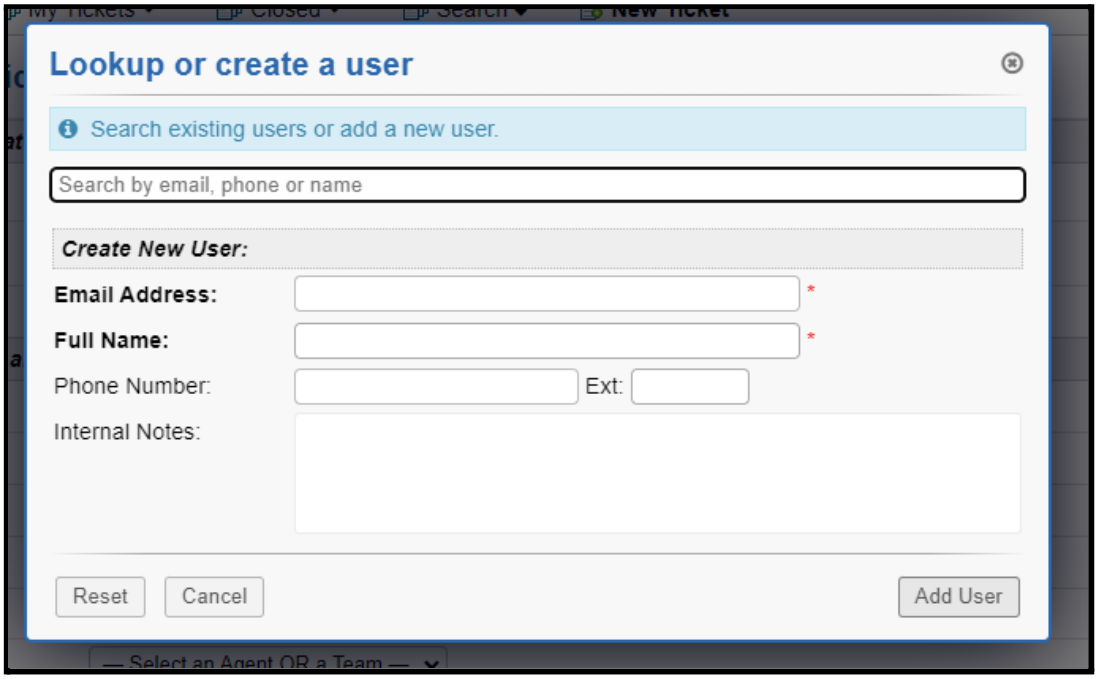
Subtotal de artículos:	**44,99 €**
Total del pedido:	**44,99 €**
Pagado:	**44,99 €**
mayo 6, 2022 a través de Transferencia bancaria directa	

Como se ha comentado anteriormente, el siguiente paso es crear el ticket en la plataforma de soporte. Para ello, el agente accede a través de esta página (https://ayuda.glez.cloud/scp/index.php) y, después de iniciar sesión con su usuario y contraseña, hacer clic en el botón “New ticket”.

Esta es la pantalla principal de osTicket:



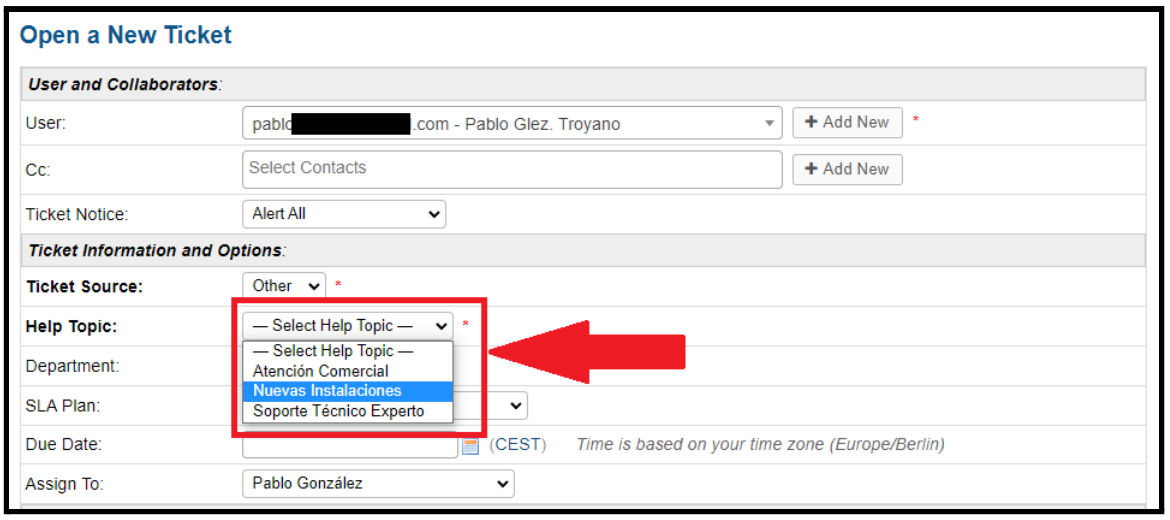
Al hacerlo, el sistema le solicita que indique el cliente para el que se abrirá el nuevo ticket. Puede usar el buscador o crear un nuevo contacto con el formulario:



En la pantalla de creación del ticket el agente debe rellenar una serie de datos:

- Además del usuario solicitante, usuarios en CC de existir la necesidad. También debe seleccionar si se enviarán mensajes de notificación de apertura del ticket.
- Fuente y categoría del ticket, así como departamento, agente y plan SLA[113]. En tanto a la categoría del ticket, se ha creado la categoría/*Help Topic* "Nuevas instalaciones" para agrupar todos los tickets de esta clase:

[113] Service Level Agreement, Acuerdo de Nivel de Servicio.

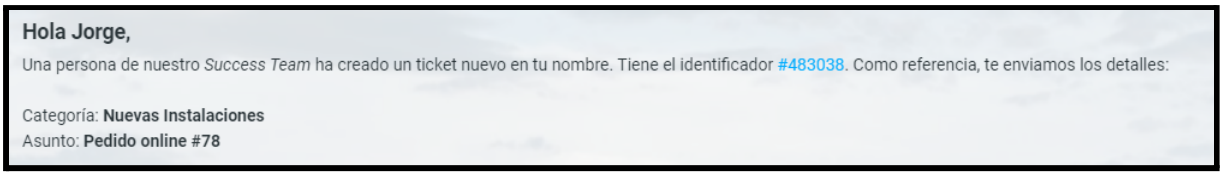


- En tanto a los detalles del ticket, el mensaje para las nuevas instalaciones debe ser un mensaje de agradecimiento y un resumen del pedido en primera instancia.

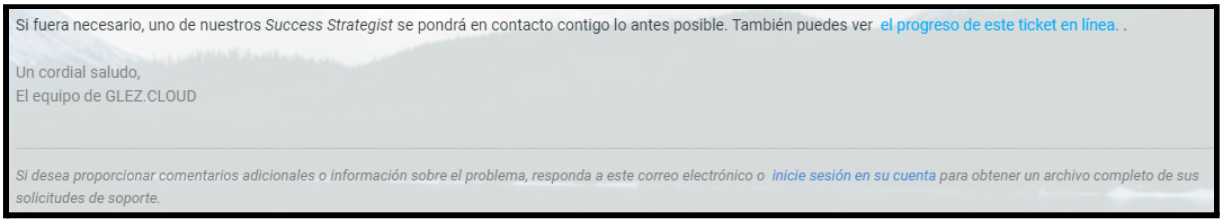
El correo que recibe el usuario es algo diferente, pues se ha configurado osTicket para que añada antes y después del mensaje del agente información adicional. Se adjunta aquí[114] un ejemplo de correo electrónico recibido. La sección que añade al

[114] https://github.com/gonzaleztroyano/ASIR2-PFC/tree/main/5-support-voip/correo-nueva-instalacion.eml.txt

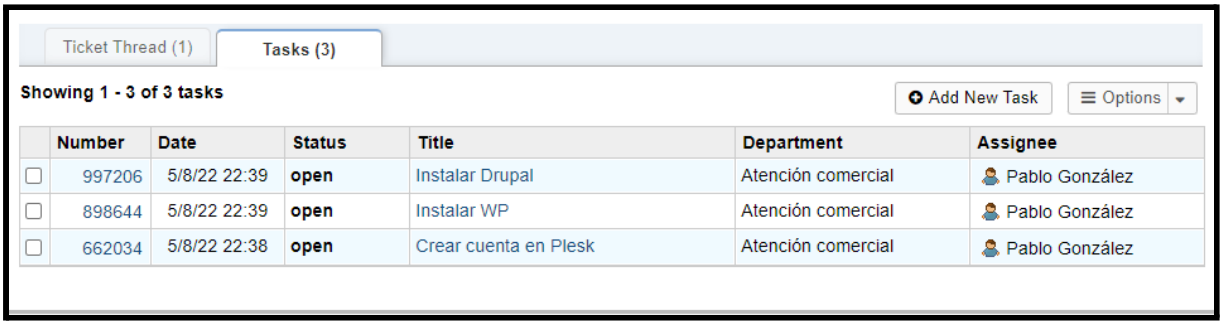
inicio del mensaje, con información sobre el identificador del nuevo ticket, el asunto y la categoría, es la siguiente:



Para finalizar el mensaje, el sistema añade la siguiente *faldilla* con un enlace a la gestión en línea del ticket a través de la web de soporte, un saludo y un recordatorio de que puede agregar comentarios adicionales respondiendo al mensaje de correo electrónico recibido.

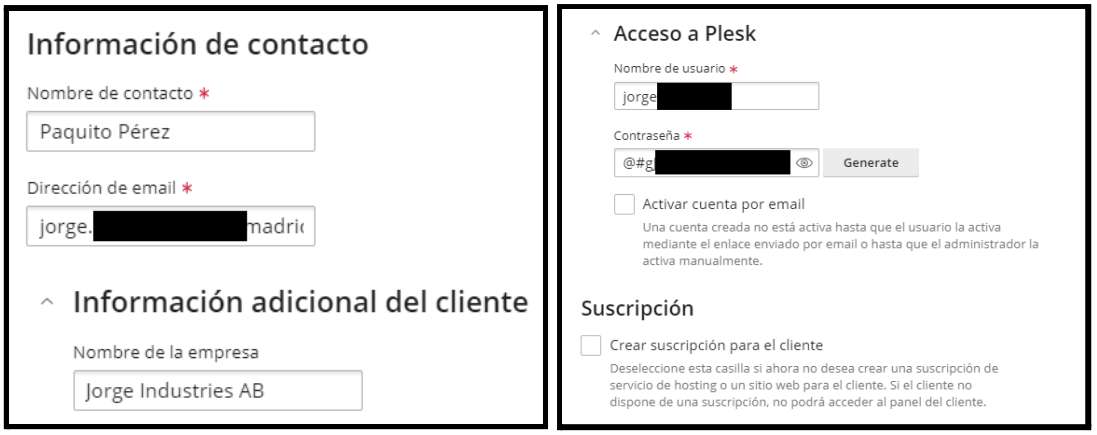


De forma interna, al crear el ticket nuevo, el agente que lo crea debe añadir tareas a este. Estas tareas se corresponderán con las acciones que son necesarias para finalizar con éxito la entrega del servicio. Las tareas pertenecen al ticket y un ticket no puede ser marcado como resuelto si tiene tareas abiertas. Se ha estudiado la posibilidad de que las tareas se creasen automáticamente al crear un ticket dentro de la categoría *Nuevas instalaciones*, pero no es posible[115] en este momento[116].

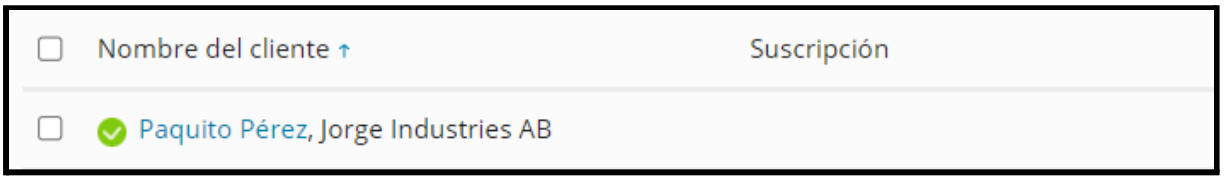


Para completar la tarea “Crear cuenta en Plesk” nos desplazamos hasta el panel de administración y aquí seguimos los pasos ya descritos para la creación de la cuenta.

[115] https://forum.osticket.com/d/100635-automatically-create-task-when-get-an-email-sent-in-osticket
[116] https://forum.osticket.com/d/93417-automation-of-task-creation



No crearemos en este momento la suscripción para hacerlo posteriormente. Al hacer clic en crear cliente, pasados unos segundos, ya lo veremos en la lista. Puesto que no hemos creado suscripción alguna hasta el momento, esta columna se encuentra vacía:

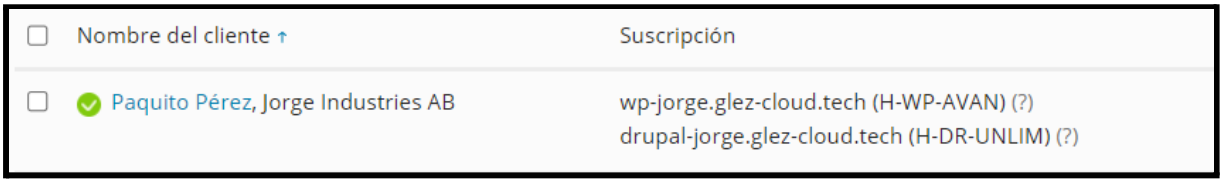


Llegados a este punto, marcamos en osTicket la tarea de *Crear cuenta en Plesk* como cerrada. Continuamos con el resto de tareas.

Para la creación de WordPress y Drupal seguimos pasos similares a los definidos en las secciones 6.5.1. Suscripción de H-WP-AVAN y 6.5.2. Suscripción de H-DR-BAS de este documento.

Una vez creadas ambas suscripciones de dominios, confirmaremos que las webs son accesibles (usando la página web pública de cada una) y enviaremos la confirmación al cliente usando el ticket de soporte abierto previamente en osTicket.

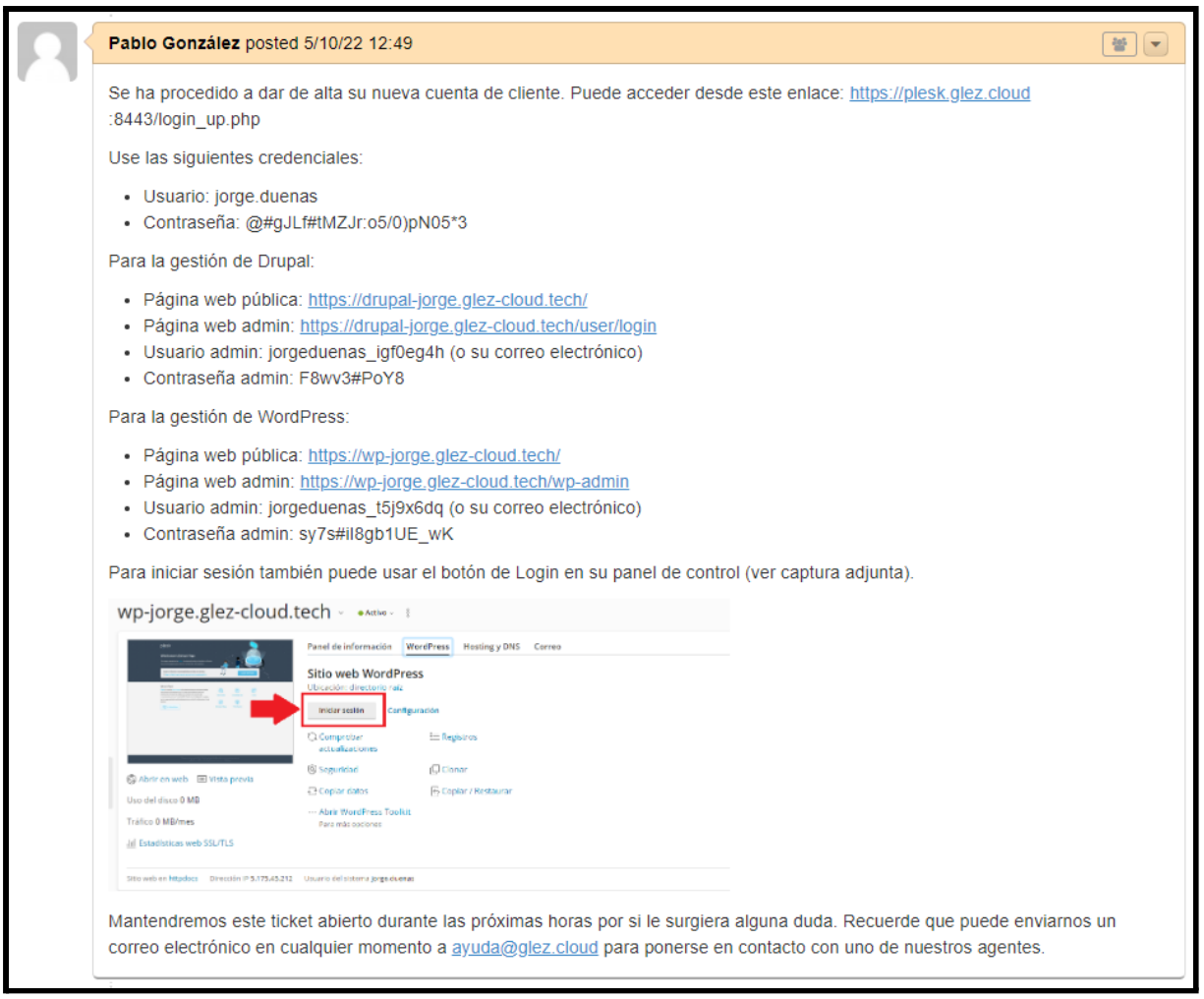
Desde el panel de Plesk veremos el cliente, ahora con ambas suscripciones disponibles y activadas:



En el mensaje enviado al cliente se incluirá toda la información sobre los nuevos servicios contratados: URLs de acceso, usuarios, contraseñas, etc.

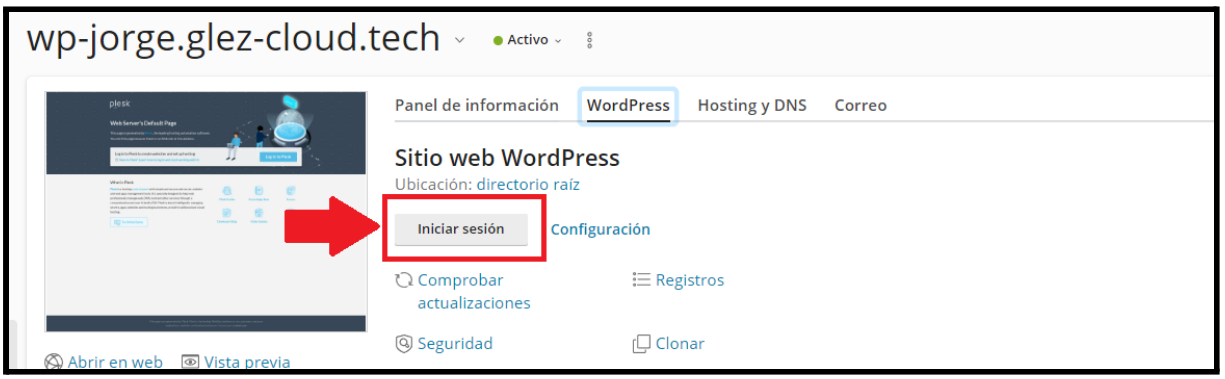
Este mensaje es importante pues contiene toda la información necesaria para que el cliente pueda comenzar a operar con sus servicios contratados. Además, si tuviera alguna duda basta con que responda al correo electrónico que ha recibido en su bandeja de entrada para que uno de los *Success Specialists* de GLEZ.CLOUD le atienda en todo lo que necesite.

Este es un ejemplo de un mensaje de confirmación post-instalación:



Si bien se incluyen las contraseñas de todos los servicios instalados, únicamente conociendo la contraseña de Plesk se puede acceder a la gestión de todos los servicios contratados. Esta información se comunica al cliente en el mensaje.

Desde la página del sitio web en Plesk se puede observar el botón de iniciar sesión en el sitio web:



7. Código y anexos

7.1. Anexo I: Dominios disponibles

7.1.1. Relación de dominios disponibles

Dominio	Uso
glez.cloud	Página pública de la empresa ficticia. Las páginas de gestión y acceso web serán subdominios de este.
glez-cloud.tech	Para evitar *ensuciar* el dominio principal, se utilizará este dominio para la infraestructura, y algunas pruebas limitadas. Es una práctica que realizan muchas empresas, como Google con su dominio `1e100.net`.
pglez.es	Acortamiento de URLs para facilitar el acceso a la documentación
pérez.es	Simulación de clientes
gómez.es	
ahorramás.es	
ahorramás.com	
carpet4you.site	
villablanca.me	Panel plesk para el apartado del Proyecto *6. Soluciones Out-of-the-box*

7.1.2. Entradas DNS ya asignadas

Las entradas de DNS del dominio glez.cloud y glez-cloud.tech, para evitar dependencias circulares *peligrosas* serán gestionadas desde el proveedor Cloudflare.

El resto de entradas DNS y servidores NS serán administrados mayoritariamente por los servicios instalados en la infraestructura desplegada como demostración del correcto funcionamiento de esta.

Para las entradas de GCP se definirán uno o varios, si existiera más de un servidor, registros A con la IP del servidor asignada por Google. Las IP en las *cloud* públicas serán siempre elásticas (estáticas, o cualquier otra nomenclatura que aplicara el proveedor de infraestructura) para evitar interrupciones en el servicio:

Las entradas A seguirán la siguiente nomenclatura:

`{GCP-VM-NAME}.gcp.glez-cloud.tech`

En Clouding.io:

`{OVH-VM-NAME}.cio.glez-cloud.tech`

En OVH Public Cloud:

`{OVH-VM-NAME}.ovh.glez-cloud.tech`

Para el resto de proveedores IaaS:
`{{PROV}-VM-NAME}.{PROV}.glez-cloud.tech`

7.2. Anexo II: Siglas y abreviaturas

Sigla/Abreviatura	Significado
A / AAAA (Registro)	Un registro tipo A (para IPv4) o AAAA (para IPv6) enlaza un nombre de dominio con una o varias direcciones IP.
API	Una Application Programming Interface, en castellano interfaz de programación de aplicaciones facilita la comunicación entre aplicaciones (o capas de ellas) usando métodos definidos.
ASN	Autonomous System Number es “un grupo de redes IP que poseen una política de rutas propia e independiente”. Véase: https://es.wikipedia.org/wiki/Sistema_autónomo
CA	Certificate Authority. En el campo de los certificados electrónicos x.509, entidad que emite y firma certificados para clientes.
CardDAV/CalDAV	Protocolos de acceso a calendario y contactos mediante, en la mayoría de los casos, clientes de correo electrónico
CLI	Command-Line Interface, en inglés. En castellano, interfaz de línea de comandos. Es la “consola”.
CMS	Content Management System. Sistema de Gestión de Contenidos. Permite publicar y gestionar contenido web sin conocimientos de programación (HTML, JS, CSS, etc). WordPress es un ejemplo de CMS
CNAME (Registro)	Es un tipo de registro que asigna un “nombre canónico”, o alias, a una dirección. Veáse: https://en.wikipedia.org/wiki/List_of_DNS_record_types
CPU	*Central Processing Unit*. Coloquialmente es el “procesador del ordenador”. En este documento se usa para referirse al uso del mismo.
DKIM	*Domain Keys Identified Mail*. Véase el Anexo IX de este documento para más información.
DMARC	*Domain-based Message Authentication Reporting and Conformance*. Véase el Anexo IX de este documento para más información.
DNS	*Domain Name System*. Entre otras funciones, se encarga de resolver la relación entre un dominio y una IP. Véase el siguiente enlace para más información:

Sigla/Abreviatura	Significado
	https://es.wikipedia.org/wiki/Sistema_de_nombres_de_dominio
DNSSEC	*Domain Name System Security Extensions*, mejoras de seguridad aplicadas al sistema DNS que añade seguridad basada en criptografía.
FQDN	*Fully Qualified Domain Name*. Nombre de dominio que identifica inequívocamente a un equipo en una red.
GCP	Google Cloud Platform, plataforma de Cloud Pública ofrecida por Google.
GUI	*Graphical User Interface*, en castellano *Interfaz Gráfica de Usuario*. Lo opuesto a CLI. En una GUI el operador/a interactúa con el sistema mediante ventanas e iconos.
HTTP/S	El Protocolo de transferencia de hipertexto (en inglés, *Hypertext Transfer Protocol*) es el protocolo mediante el cual se transmiten las páginas web. La “S” se añade a las conexiones cifradas entre el cliente y el servidor.
IdP	*Identity Provider*. Proveedor de Identidad. Es el agente o servicio encargado de asegurar la identidad de un recurso o usuario.
IMAP	*Internet Message Access Protocol*. En castellano, *Protocolo de Acceso a Mensajes en Internet.* Permite a los clientes acceder a su correo electrónico acceder a los mensajes de correo electrónico de una cuenta.
IOPS	*Input/Output Operations Per Second*. En castellano, Operaciones de Entrada/Salida por segundo.
IP (dirección)	Pudiendo ser IPv4 e IPv6, es el identificador de un equipo en una red dada.
IPv4	Dirección de un equipo en una red. Se usan 32 bits, separados en 4 octetos. Ejemplo: 221.125.255.13
IPv6	Dirección de un equipo en una red. Se usan 128 bits, separados en 8 grupos de 16 bits.
IT	*Information Technology*.
IVR	Literalmente es *Interactive Voice Response*. Pero se conoce como “IVR” a la práctica mayoría de los menús que permiten que el usuario interactúe con una serie de opciones mediante una llamada de teléfono.

Sigla/Abreviatura	Significado
JSON	*JavaScript Object Notation*. En castellano, *notación de objeto de JavaScript*. Formato de intercambio de datos en auge gracias a su versatilidad.
MFA	*Multi-Factor Authentication*. Autenticación Multifactor, en castellano. Además de la contraseña es necesario un segundo control para poder iniciar sesión en un sistema.
MUA / MTA	*Mail User Agent* y *Mail Transfer Agent*. Tipo de software que realizan acciones de transferencia en el envío de mensajes de correo electrónico. Bien servidor-cliente o bien servidor-servidor, respectivamente.
MV / VM	*Máquina Virtual* / *Virtual Machine*
OSI	*Open Systems Interconnection*. Es un modelo teórico de referencia para la arquitectura de redes. Se divide en 7 niveles: 1 (físico) - 7 (Aplicación)
POP3	*Post Office Protocol version 3*. Protocolo que permite a los clientes acceder a su correo electrónico acceder a los mensajes de correo electrónico de una cuenta.
pps	*packet per second*. Paquetes por segundo. Unidad de medida para la transferencia en una red.
RAM	*Random Access Memory*. Memoria de Acceso Aleatorio. En esta se alojan los datos de las aplicaciones en funcionamiento.
RFC	*Request For Comments*. En castellano, *Petición de comentarios*. Es la forma que tiene la comunidad técnica de crear y publicar protocolos.
RIPE	Las siglas correctas son “RIPE NCC”, aunque no es común usarla. “RIPE” es usado. Son las siglas de *Réseaux IP Européens Network Coordination Centre*. Es el RIR para Europa, Oriente Medio y parte de Asia. Véase el siguiente enlace: https://www.ripe.net/about-us/what-we-do
SEO	*Search Engine Optimization*. En castellano, *Optimización para motores de búsqueda*. Es la técnica que intenta posicionar una web lo más alto en los resultados de un buscador. Se suele combinar con SEM (tráfico de pago, anuncios, en los motores de búsqueda).
SFTP	*Secure File Transfer Protocol*. En castellano*, Protocolo de transferencia segura de archivos*. Es una evolución de FTP

Sigla/Abreviatura	Significado
	(no seguro).
SIP	*Session Initiation Protocol*- En castellano, *Protocolo de iniciación de sesión*. Se usa en el mundo de VoIP para iniciar, señalizar, mantener y cerrar una sesión entre 2 o más participantes.
SMTP	*Simple Mail Transfer Protocol*. En castellano, protocolo para transferencia simple de correo, permite el intercambio de mensajes de correo electrónico entre varios equipos servidores.
SPF	*Sender Policy Framework*. Véase el Anexo IX de este documento para más información.
SSH	*Secure SHell*. Es un protocolo que permite la administración (y la conexión general gracias al establecimiento de túneles) remota de un equipo. Usa conexiones cifradas basadas en certificado.
TCP	*Transmission Control Protocol*. Es un protocolo a nivel 3 de OSI que asegura la entrega de los distintos segmentos.
TLS / SSL	*Transport Layer Security* y Secure Sockets Layer (antiguo) permiten el establecimiento de conexiones seguras en una red.
TXT (Registro)	Tipo de Registro de Recurso que permite añadir información adicional a un nombre DNS.
UDP	*User Datagram Protocol*. Es junto a TCP un protocolo a nivel 3 de OSI. En contraposición con este, UDP es más liviano y no asegura la entrega de todos los segmentos/paquetes.
UX	*User eXperience*. Técnica/Ciencia/Área que estudia la experiencia de los usuarios con interfaces, operaciones y protocolos. Su máxima es asegurar una experiencia cómoda y agradable.
VoIP	*Voice over Internet Protocol*. Permite el envío de voz sobre redes IP.
YAML	Lenguaje de marcado, parecido a XML aunque más ligero.

7.3. Anexo III: Contraseñas de los servicios

URL/Servicio	usuario	Contraseña
https://manage-portainer.glez.cloud:9000	pablo_1vmSzOqMab8s	gwyHHLqv$B*z4M1U
https://manage-dns.glez.cloud	pablo_lvg5OU7RyAt7	S#rtVzv9HdM!n@XMemV^W
https://manage-dns.glez.cloud	ahorramassa_segismundo	5fv8riEH*ibF4dtc
https://mail.glez.cloud/	pablo_ucacjtaetcj5	Q^xo^Ev*J%yqfmPi
https://mail.glez.cloud/	pablo@carpet4you.site	=,iecArSIt,69
https://mail.glez.cloud/ (App Password)	pablo@carpet4you.site	80,},SPLaimastItErypROPINaPHOBlEthusE
https://mail.glez.cloud/SOGo/	ricardo@glez-cloud.tech	94,&,siTIcIan
https://tienda.demodocu1.villablanca.me/admin	demodocu1	+n3s63pZRbkABh28
https://glez-cloud.vservers.es:8443/	axarnet@glez.cloud	u#2162jUo
SSH://glez-cloud.vservers.es	root	RHjQsp4M$
(Plesk) ahorramás.es	ahorramas.es_vu1pvzwkfi	W8TI5_ync1
	axarnet_njaoblja axarnet@glez.cloud	wVzI4D@%0wbz6ylX
Plesk - Test-User-1YDNS - 2 Zonas + Correo	Test-User-1YDNS@glez.cloud	S?7Dw_sM2w7mxavt
Plesk - Test-User-JoomlaAVAN	Test-User-JoomlaAVAN@glez.cloud	O@0UpfwH1a^c2dgm
joomla-avan.glez-cloud.tech	Test-User-JoomlaAVAN@glez.cloud -- Test-User-JoomlaAVAN_dpdz0922	5E0iLh!m4T
Plesk - Test-User-WP-AVAN	Test-User-WP-AVAN	s!ng8dN?2NXk9lwq

drupal-bas.glez-cloud.tech	Test-User-WP-AVAN_dlvc2u62 Test-User-WP-AVAN@glez.cloud	t04fE2K_pW
Plesk - test-paquito-perez(@glez.tk)	test-paquito-perez	tKb4ax@7FD4?anrp
Joomla - drupal-test-paquito-perez.glez-cloud.tech	test-paquito-perez_u1236yiq test-paquito-perez@glez.tk	86v#Hs7MuD
WEBMAIL - AYUDA	ayuda@glez.cloud	y:#NdOwEE5}Ba0}#_k.;42+0
AYUDA - Google - App Password		lxrllcvgcsprcnob
osTICKET - pablo	pablo -- pablo@glez.cloud	/XIM@3&4Mds.RYw$>{zxoA%0
Plesk - Admin Visitante	visitante-admin (o visitante-admin@glez.cloud)	|t=#!M)3)}QJuj1eoj.6l_uE
osTicket - Admin Visitante	visitante-admin (o visitante-admin@glez.cloud)	Q+K%7n6j~]&h1f[(0:8!RGj)
FreePBX Admin Panel http://27.0.173.142/admin/config.php	admin489318357321169043373080	QB76;>wKxjv)CyIb3s5!wadn
FreePBX SIP Phone	901@voip.glez.cloud	901
FreePBX SIP Phone	902@voip.glez.cloud	902
FreePBX SIP Phone	903@voip.glez.cloud	e01ff51487d8b1bcf27074373525db70

7.4. Anexo IV: Códigos relativos al servicio DNS

7.4.1. Creación de tablas SQLITE

```
PRAGMA foreign_keys = 1;

CREATE TABLE domains (
  id                    INTEGER PRIMARY KEY,
  name                  VARCHAR(255) NOT NULL COLLATE NOCASE,
  master                VARCHAR(128) DEFAULT NULL,
  last_check            INTEGER DEFAULT NULL,
  type                  VARCHAR(6) NOT NULL,
  notified_serial       INTEGER DEFAULT NULL,
  account               VARCHAR(40) DEFAULT NULL
);

CREATE UNIQUE INDEX name_index ON domains(name);


CREATE TABLE records (
  id                    INTEGER PRIMARY KEY,
  domain_id             INTEGER DEFAULT NULL,
  name                  VARCHAR(255) DEFAULT NULL,
  type                  VARCHAR(10) DEFAULT NULL,
  content               VARCHAR(65535) DEFAULT NULL,
  ttl                   INTEGER DEFAULT NULL,
  prio                  INTEGER DEFAULT NULL,
  change_date           INTEGER DEFAULT NULL,
  disabled              BOOLEAN DEFAULT 0,
  ordername             VARCHAR(255),
  auth                  BOOL DEFAULT 1,
  FOREIGN KEY(domain_id) REFERENCES domains(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE INDEX rec_name_index ON records(name);
CREATE INDEX nametype_index ON records(name,type);
CREATE INDEX domain_id ON records(domain_id);
CREATE INDEX orderindex ON records(ordername);


CREATE TABLE supermasters (
  ip                    VARCHAR(64) NOT NULL,
  nameserver            VARCHAR(255) NOT NULL COLLATE NOCASE,
  account               VARCHAR(40) NOT NULL
);

CREATE UNIQUE INDEX ip_nameserver_pk ON supermasters(ip, nameserver);


CREATE TABLE comments (
  id                    INTEGER PRIMARY KEY,
  domain_id             INTEGER NOT NULL,
  name                  VARCHAR(255) NOT NULL,
  type                  VARCHAR(10) NOT NULL,
```

```
  modified_at           INT NOT NULL,
  account               VARCHAR(40) DEFAULT NULL,
  comment               VARCHAR(65535) NOT NULL,
  FOREIGN KEY(domain_id) REFERENCES domains(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE INDEX comments_domain_id_index ON comments (domain_id);
CREATE INDEX comments_nametype_index ON comments (name, type);
CREATE INDEX comments_order_idx ON comments (domain_id, modified_at);


CREATE TABLE domainmetadata (
 id                     INTEGER PRIMARY KEY,
 domain_id              INT NOT NULL,
 kind                   VARCHAR(32) COLLATE NOCASE,
 content                TEXT,
 FOREIGN KEY(domain_id) REFERENCES domains(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE INDEX domainmetaidindex ON domainmetadata(domain_id);


CREATE TABLE cryptokeys (
 id                     INTEGER PRIMARY KEY,
 domain_id              INT NOT NULL,
 flags                  INT NOT NULL,
 active                 BOOL,
 content                TEXT,
 FOREIGN KEY(domain_id) REFERENCES domains(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE INDEX domainidindex ON cryptokeys(domain_id);


CREATE TABLE tsigkeys (
 id                     INTEGER PRIMARY KEY,
 name                   VARCHAR(255) COLLATE NOCASE,
 algorithm              VARCHAR(50) COLLATE NOCASE,
 secret                 VARCHAR(255)
);

CREATE UNIQUE INDEX namealgoindex ON tsigkeys(name, algorithm);
```

7.4.2. Archivo de configuración */etc/powerdns/pdns.conf*

```
api=yes
api-key=JmnWB4iiphR6FyzygJ3sdrx1u5OCas
api-logfile=/var/log/pdns.log
#######
webserver=yes
webserver-address=0.0.0.0
webserver-allow-from=0.0.0.0/0
webserver-port=8081
#######
setgid=pdns
setuid=pdns
#######
launch=gsqlite3
gsqlite3-database=/var/lib/powerdns/pdns.sqlite3
# Hasta aquí actualizado el 13/03/2022

# Actualizado a continuación de esta línea en fecha 14/03/2022
loglevel=6
query-logging=yes
log-dns-details=yes
log-dns-queries=yes
logging-facility=0
disable-syslog=no
```

7.5. Anexo V: Respecto a la nomenclatura punycode

Tal y como se puede leer en este artículo en la Wikipedia[117], es una sintaxis de codificación usada en programación que usa una cadena Unicode que puede ser traducida en una cadena de caracteres más limitada compatible con los nombres de red. La documentación técnica está recogida en el documento *Request for Comments* (RFC) número 3492, accesible desde este enlace[118].

En bajo nivel, esto es muy útil para los nombres de dominios puesto que por implementación solo aceptan caracteres UTF-8. Pero los caracteres internacionales no los podemos codificar así. Por ejemplo, ñ, ç, á, à, â, ä, etc.

Veamos algún ejemplo en la siguiente tabla:

UTF-8	Punycode
españa.es	xn--espaa-rta.es
ahorramás.com	xn--ahorrams-fza.com
pérez.es	xn--prez-bpa.es

Podemos utilizar este conversor[119] online para convertir nombres de dominio entre punycode y unicode. Aunque se aplica sobre todo en nombres de dominio, no es exclusivo. Si bien su implementación en otros campos es muy limitada. Si fuéramos a enviar un correo electrónico a un dominio con caracteres especiales es posible que tengamos que hacer la conversión antes.

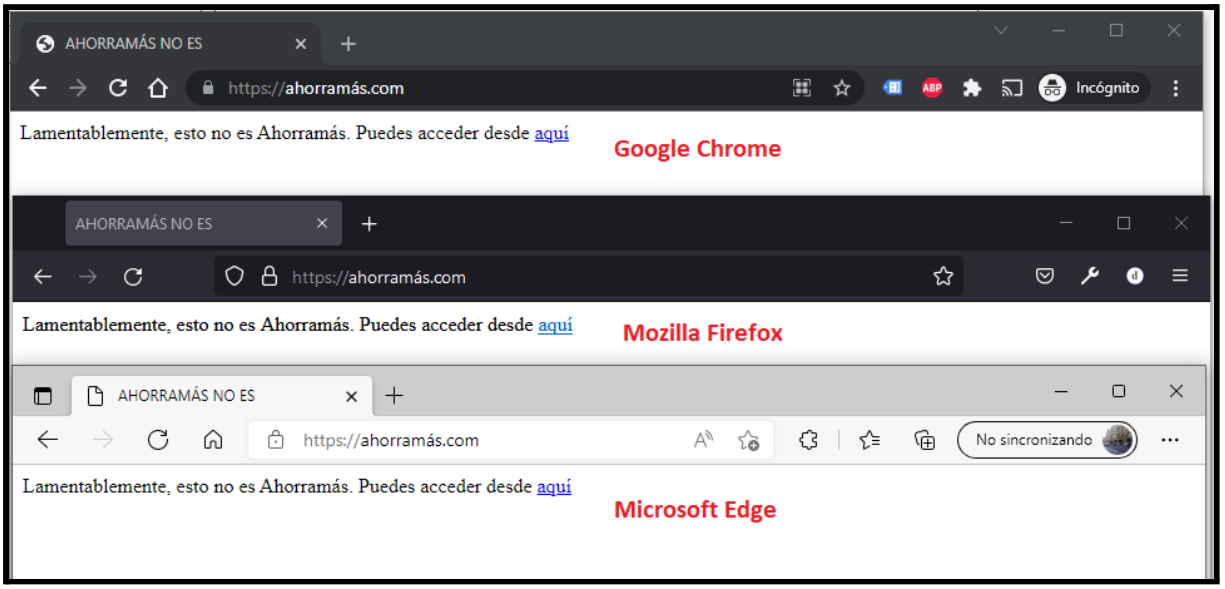
Sin embargo, *no es oro todo lo que reluce* en esto de la implementación Punycode. “Los malos” también han encontrado formas de aprovechar esta innovación para hacerse pasar por bancos, redes sociales y demás actores legítimos.

Perfectamente se podría haber hecho con el dominio de ahorramás.com. Por supuesto, sobra decir que no ha sido mi caso.

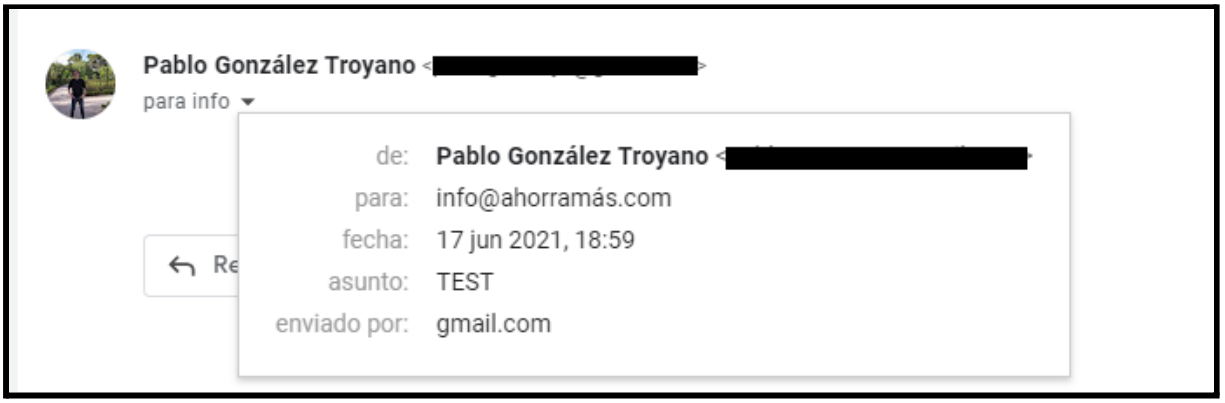
[117] https://es.wikipedia.org/wiki/Punycode
[118] https://datatracker.ietf.org/doc/html/rfc3492/
[119] https://punycode.es/

Sin embargo, cabe destacar que en los navegadores web muchas personas no notarían ninguna diferencia y pueden no darse cuenta de si están (o no) navegando por el sitio que realmente creen estar visitando.

Así es como se ve el dominio ahorramás.com (`https://xn--ahorrams-fza.com/`) en los distintos navegadores web (Google Chrome, Mozilla Firefox y Microsoft Edge):



Tampoco es detectado fácilmente en los correos electrónicos:



Hay multitud de publicaciones técnicas[120] [121] [122] tratando este tema. Sin duda, muy importante del lado de la ciberseguridad.

[120] https://www.elladodelmal.com/2014/10/ataques-de-phishing-con-codigos.html
[121] https://unaaldia.hispasec.com/2021/10/punycode-es-utilizado-en-ataques-a-traves-de-google-ads-para-distribuir-malware.html
[122] https://ui.adsabs.harvard.edu/abs/2020arXiv200613742S/abstract

7.6. Anexo VI: Códigos relativos al servicio de correo

7.6.1. Archivo de configuración de mailcow

```
MAILCOW_HOSTNAME=mail.glez.cloud
MAILCOW_PASS_SCHEME=BLF-CRYPT
DBNAME=mailcow
DBUSER=mailcow
DBPASS=JELM312L6TRuY6T1fv6w2kWnWbs5
DBROOT=3LfWpG5Zuo8vmWgPK5llxpgXbTyj
HTTP_PORT=80
HTTP_BIND=
HTTPS_PORT=443
HTTPS_BIND=
SMTP_PORT=25
SMTPS_PORT=465
SUBMISSION_PORT=587
IMAP_PORT=143
IMAPS_PORT=993
POP_PORT=110
POPS_PORT=995
SIEVE_PORT=4190
DOVEADM_PORT=127.0.0.1:19991
SQL_PORT=127.0.0.1:13306
SOLR_PORT=127.0.0.1:18983
REDIS_PORT=127.0.0.1:7654
TZ=Europe/Madrid
COMPOSE_PROJECT_NAME=mailcowdockerized
ACL_ANYONE=disallow
MAILDIR_GC_TIME=7200
ADDITIONAL_SAN=
ADDITIONAL_SERVER_NAMES=
SKIP_LETS_ENCRYPT=n
ENABLE_SSL_SNI=n
SKIP_IP_CHECK=n
SKIP_HTTP_VERIFICATION=n
SKIP_CLAMD=n
SKIP_SOGO=n
SKIP_SOLR=n
SOLR_HEAP=1024
ALLOW_ADMIN_EMAIL_LOGIN=n
USE_WATCHDOG=y
WATCHDOG_NOTIFY_BAN=n
WATCHDOG_EXTERNAL_CHECKS=n
WATCHDOG_VERBOSE=n
LOG_LINES=9999
IPV4_NETWORK=172.22.1
IPV6_NETWORK=fd4d:6169:6c63:6f77::/64
```

```
MAILDIR_SUB=Maildir
SOGO_EXPIRE_SESSION=480
DOVECOT_MASTER_USER=
DOVECOT_MASTER_PASS=
ACME_CONTACT=
WEBAUTHN_ONLY_TRUSTED_VENDORS=n
```

7.6.2. Cabeceras de mensajes de correo

cabeceras-1.eml.txt

```
Delivered-To: [REDACTED]^^@^^gonzaleztroyano.es
Received: by 2002:a05:6838:f30f:0:0:0:0 with SMTP id z15csp1845841nki;
        Mon, 21 Mar 2022 11:34:30 -0700 (PDT)
X-Google-Smtp-Source: ABdhPJywmWEb/LBtIQx+LUWCRCz4ntqTjXnhzB+HMgdipUhMnmG3TIqvuRyL4+2mPqbIu6DFOJ7V
X-Received: by 2002:a05:600c:4e0f:b0:38c:b6d9:511b with SMTP id b15-20020a05600c4e0f00b0038cb6d9511bmr124822wmq.96.1647887669934;
        Mon, 21 Mar 2022 11:34:29 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1647887669; cv=none;
        d=google.com; s=arc-20160816;
        b=pZIyeiVNpEqZl8PUzEcqDdh2sxz+0phiG1hbk/Flr8ZWJ17UhWaisv3xzP1GHxvmbf
         JEk5UCinVVm7pEe6FYakYT5/b1is6st4uBUWWuDszselvTFZWHMcrV1Jm0TANjMHLP0/
         95v28YFOvnYKp1cOtSGxBgiKsrOXGeJugFsmPAqII0626b+VHg5O16OJDwyqsyLXkoHO
         iSH/x8J08PptccxxuPk2fgAbvWxStHAxF3+PYzHITHMkWvnqz7hfzcnq6qS/ET6F24Xb
         u3fEN9kObX1OpyP7PGZwncHEAWmPRQjvt3JnjvWpksZ6Fas1lIB6W8kD6oG8HHHqoNnO
         scJQ==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;
        h=message-id:subject:date:mime-version:user-agent:to:from
         :dkim-signature;
        bh=ydTvrRvaL6epW+mSA2PptNZuH6FPXFIyicuu785ZKAA=;
        b=NqjO3uvjSmWIWZtzg+bG4nMo34BUPYWQJWwXcwxCvJxb+D5hw9PH4OyCkhGqmky/Az
         gE+0iORFMVxobZrTjhXtfmyGfCu0JDokJYhc5vu3byeENMC2QMugqnFclrtbHnnM9sI5
         wHAmcyvijpSa52oO63BqAQw5Smt8ZdtOu2oUHRkKOe7mjWXk5BrMBGpeUU2XCQdZzHtW
         qUg/u0naXlXGxj9ET+EEl2rcXLyv4CI7BA3xnpxqDE+oUDYI5C53V2VB/mTFRN4Lo9hG
         8m5SrdUX+FGZCofuJfvcPDiJF1ppGzE5OpRDdEE4bvd0tWMTckkiUe4j7UaT+0QhJPFQ
         QmzA==
ARC-Authentication-Results: i=1; mx.google.com;
       dkim=pass header.i=@carpet4you.site header.s=glezcloud header.b=DemdnIGl;
       spf=pass (google.com: domain of pablo^^@^^carpet4you.site designates 93.189.91.9 as permitted sender)
smtp.mailfrom=pablo^^@^^carpet4you.site
Return-Path: <pablo^^@^^carpet4you.site>
Received: from mail.glez.cloud (de2afb89-06d4-4df5-a344-0d24c913351e.clouding.host. [93.189.91.9])
        by mx.google.com with ESMTPS id w9-20020a5d6089000000b00203e9019308si7283664wrt.140.2022.03.21.11.34.29
        for <[REDACTED]^^@^^gonzaleztroyano.es>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Mon, 21 Mar 2022 11:34:29 -0700 (PDT)
Received-SPF: pass (google.com: domain of pablo^^@^^carpet4you.site designates 93.189.91.9 as permitted sender)
client-ip=93.189.91.9;
Authentication-Results: mx.google.com;
       dkim=pass header.i=@carpet4you.site header.s=glezcloud header.b=DemdnIGl;
       spf=pass (google.com: domain of pablo^^@^^carpet4you.site designates 93.189.91.9 as permitted sender)
smtp.mailfrom=pablo^^@^^carpet4you.site
Received: from [127.0.0.1] (localhost [127.0.0.1]) by localhost (Mailerdaemon) with ESMTPA id E42436E804
          for <[REDACTED]^^@^^gonzaleztroyano.es>; Mon, 21 Mar 2022 19:34:28 +0100 (CET)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=carpet4you.site;
          s=glezcloud; t=1647887669;
          h=from:subject:date:message-id:to:mime-version:content-type;
          bh=ydTvrRvaL6epW+mSA2PptNZuH6FPXFIyicuu785ZKAA=;
          b=DemdnIGlvNPVo0LdEKCumDUV7JF0efDRoPxrWOvm9gJN8hJKyLIGrYUA4qUS4nYrjx7V09
          Ucruh1U4Ef8upYIl6hrjqn8u3q9Sz0g6XK4+fQk774Z6afrCU6TZfB3hOx6az5F0L6vk0B
          noSa0ftck6Nh73dqCgabBdqVPvzgcdQ=
From: =?utf-8?q?Pablo_Gonz=C3=A1lez?= <pablo^^@^^carpet4you.site>
To: [REDACTED]^^@^^gonzaleztroyano.es
User-Agent: SOGoMail 5.5.0
MIME-Version: 1.0
Date: Mon, 21 Mar 2022 19:34:28 +0100
Subject: TEST de =?utf-8?q?env=C3=ADo?= desde Mailcow
Message-ID: <3c-6238c500-7-672a0300@214315766>
Content-Type: multipart/alternative; boundary="----=_=-_OpenGroupware_org_NGMime-60-1647887668.466402-1------"
X-Last-TLS-Session-Version: None

------=_=-_OpenGroupware_org_NGMime-60-1647887668.466402-1------
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 115


=C2=A1Hola!
```

```
Esto es un mensaje de prueba sin mayor importancia. =C2=A1Veamos como l=
lega!

Un saludito,
Pablito

------=_=-_OpenGroupware_org_NGMime-60-1647887668.466402-1------
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 155

<html>=C2=A1Hola!<br /><br />Esto es un mensaje de prueba sin mayor imp=
ortancia. =C2=A1Veamos como llega!<br /><br />Un saludito,<br />Pablito=
</html>

------=_=-_OpenGroupware_org_NGMime-60-1647887668.466402-1--------
```

cabeceras-2.eml.txt

```
Return-Path: <pablo.[REDACTED]*^^@^^educa.madrid.org>
Delivered-To: pablo[RED]*^^@^^carpet4you.site
Received: from mail.glez.cloud ([172.22.1.253])
          by 724f009ff4cf with LMTP
          id KO/KLMPHNWIoDwAAuXvktg
          (envelope-from <pablo.[REDACTED]*^^@^^educa.madrid.org>)
          for <pablo[RED]*^^@^^carpet4you.site>; Sat, 19 Mar 2022 13:08:35 +0100
Received: from mx02.puc.rediris.es (outbound2sev.lav.puc.rediris.es [130.206.19.171])
          (using TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256 bits)
           key-exchange X25519 server-signature RSA-PSS (4096 bits) server-digest SHA256)
          (No client certificate requested)
          by mail.glez.cloud (Postcow) with ESMTPS id 217F36E8E4
          for <pablo[RED]*^^@^^carpet4you.site>; Sat, 19 Mar 2022 13:08:35 +0100 (CET)
Received: from smtp.educa.madrid.org ([193.146.123.99])
          by mx02.puc.rediris.es  with ESMTP id 22JC8VaV017481-22JC8VaW017481
          for <pablo[RED]*^^@^^carpet4you.site>; Sat, 19 Mar 2022 13:08:31 +0100
Received: (qmail 26008 invoked from network); 19 Mar 2022 12:08:31 -0000
Received: from unknown (HELO WEBMAIL) ([172.16.2.201])
          (envelope-sender <pablo.[REDACTED]*^^@^^educa.madrid.org>)
          by 0 (qmail-ldap-1.03) with SMTP
          for <pablo[RED]*^^@^^carpet4you.site>; 19 Mar 2022 12:08:31 -0000
MIME-Version: 1.0
Date: Sat, 19 Mar 2022 13:08:31 +0100
From: =?UTF-8?Q?Pablo_Gonz=C3=A1lez_Troyano?=
 <pablo.[REDACTED]*^^@^^educa.madrid.org>
To: Pablo <pablo[RED]*^^@^^carpet4you.site>
Subject: Fwd: TEST desde EducaMadrid
In-Reply-To: <2ad0254ad0f21f91ba33c0ce5b8e9b50@educa.madrid.org>
References: <2ad0254ad0f21f91ba33c0ce5b8e9b50@educa.madrid.org>
User-Agent: CorreoWeb EducaMadrid
Message-ID: <b56252f6560ba2b990199de382e11fca@educa.madrid.org>
X-Sender: pablo.[REDACTED]*^^@^^educa.madrid.org
Organization: Comunidad de Madrid. EducaMadrid.
X-Remote-Browser: Mozilla/5.0 (X11; CrOS x86_64 14469.41.0) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/99.0.4844.57 Safari/537.36
X-Originating-IP: [79.116.7.222]
X-Webmail-Server: 172.16.2.201
Content-Type: multipart/alternative;
 boundary="=_c927eca8e881b2dbd59297847775eb5c"
X-FE-Policy-ID: 23:14:2:educa.madrid.org
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; d=educa.madrid.org; s=dkim_educamadrid; c=relaxed/relaxed;
 h=mime-version:date:from:to:subject:references:message-id:content-type;
 bh=de4aaNZrbBqOanRa45Nz5K2tGmAbMzk5nCoqxoXbEho=;
 b=AYkzUrnphGScZdF7iA6cApC2bIPwIztNJOF73fDzyQHohgZAHnZkG23tGECQrajTXvs461axYFRW
           9gSuL9QQ1E7Q11nmxaUDvC3Fsdny0Y1tjZ8/kjvoS5oekBd9+kZe5TXc+KFNVsvQakKZlmEbkzD0
           tVHRB1Gjnd6l5onSLKM1pdNUhkLXFC+SgVSH+M/EFusEnwvnRVkaPIodYSqx+3OLSxdtFuf3EHCd
           Yt68XQtJHEw2ighP6Rnii+CImYihkxeLbHUo7CGAX1ipvB5OUO+RbheYJVaokwV29afda3NZKV60
           pfOJKS44ZGcaUvrwK2lDvlFV6fR3kPMGL5sEjA==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=carpet4you.site;
           s=glezcloud; t=1647691715;
           h=from:from:reply-to:subject:subject:date:date:message-id:message-id:
            to:to:cc:mime-version:mime-version:content-type:content-type:
            in-reply-to:in-reply-to:references:references:dkim-signature;
           bh=de4aaNZrbBqOanRa45Nz5K2tGmAbMzk5nCoqxoXbEho=;
           b=D/840mXK3aG+xqOA/enS+N5j/ddr2KIB4T34+FeKWje9/QM0nK6LOYuqTPbZ8cc8L+ULl3
           uuF/eQsyKRHyihp5d3d7OxBFOJDsVzMbU1ik2mAdTEmmnzsexnJj5QUYjsVL08anb10I3C
           XAmrJpux6XZl0IDK+4UdNmEXu1lToOw=
ARC-Seal: i=1; s=glezcloud; d=carpet4you.site; t=1647691715; a=rsa-sha256;
           cv=none;
           b=KNwXUl6SBByUxgWWw6bbzcz1TPvQjh/Hx48Vv86lwEwXtzX/LXIQYFPOm16adbw6NTrB5v
           LfmXtlK8E7mz9pgKQNxSX0eKV9VrPNz8aRrYSgV7at9Y608d/p3hYMdQhQ4+Yp6FoVQUXm
           6DGC7niSYkwXdSzUpD4e5KCz1Ch4ST0=
ARC-Authentication-Results: i=1;
```

```
          mail.glez.cloud;
          dkim=pass header.d=educa.madrid.org header.s=dkim_educamadrid header.b=AYkzUrnp;
          dmarc=pass (policy=reject) header.from=madrid.org;
          spf=pass (mail.glez.cloud: domain of pablo.[REDACTED]*^^@^^educa.madrid.org designates 130.206.19.171 as permitted sender)
smtp.mailfrom=pablo.[REDACTED]*^^@^^educa.madrid.org
X-Last-TLS-Session-Version: TLSv1.3
X-Spamd-Result: default: False [-2.00 / 15.00];
          DWL_DNSWL_LOW(-1.00)[madrid.org:dkim];
          DMARC_POLICY_ALLOW(-0.50)[madrid.org,reject];
          R_SPF_ALLOW(-0.20)[+ip4:130.206.19.0/24:c];
          R_DKIM_ALLOW(-0.20)[educa.madrid.org:s=dkim_educamadrid];
          MIME_GOOD(-0.10)[multipart/alternative,text/plain];
          MX_GOOD(-0.01)[];
          XM_UA_NO_VERSION(0.01)[];
          BCC(0.00)[];
          HAS_XOIP(0.00)[];
          FROM_HAS_DN(0.00)[];
          TO_MATCH_ENVRCPT_ALL(0.00)[];
          ARC_NA(0.00)[];
          RCPT_COUNT_ONE(0.00)[1];
          PREVIOUSLY_DELIVERED(0.00)[pablo[RED]*^^@^^carpet4you.site];
          RCPT_MAILCOW_DOMAIN(0.00)[carpet4you.site];
          RCVD_TLS_LAST(0.00)[];
          ASN(0.00)[asn:766, ipnet:130.206.0.0/16, country:ES];
          RCVD_COUNT_THREE(0.00)[3];
          HAS_ORG_HEADER(0.00)[];
          ARC_SIGNED(0.00)[carpet4you.site:s=glezcloud:i=1];
          MID_RHS_MATCH_FROM(0.00)[];
          TO_DN_ALL(0.00)[];
          FROM_EQ_ENVFROM(0.00)[];
          DKIM_TRACE(0.00)[educa.madrid.org:+];
          MIME_TRACE(0.00)[0:+,1:+,2:~];
          RWL_MAILSPIKE_VERYGOOD(0.00)[130.206.19.171:from]
X-Rspamd-Queue-Id: 217F36E8E4
Authentication-Results: mail.glez.cloud;
          dkim=pass header.d=educa.madrid.org header.s=dkim_educamadrid header.b=AYkzUrnp;
          dmarc=pass (policy=reject) header.from=madrid.org;
          spf=pass (mail.glez.cloud: domain of pablo.[REDACTED]*^^@^^educa.madrid.org designates 130.206.19.171 as permitted sender)
smtp.mailfrom=pablo.[REDACTED]*^^@^^educa.madrid.org

--=_c927eca8e881b2dbd59297847775eb5c
Content-Transfer-Encoding: 8bit
Content-Type: text/plain; charset=UTF-8;
 format=flowed

Hola desde EducaMadrid

---
PABLO GONZÃ LEZ TROYANO
pablo.[REDACTED]*^^@^^educa.madrid.org
2Âº ASIR
IES Villablanca, Madrid
--=_c927eca8e881b2dbd59297847775eb5c
Content-Transfer-Encoding: quoted-printable
Content-Type: text/html; charset=UTF-8

<html><head><meta http-equiv=3D"Content-Type" content=3D"text/html; charset=
=3DUTF-8" /></head><body style=3D'font-size: 10pt; font-family: Arial,Helve=
tica,sans-serif'>
<p><br /></p>
<div id=3D"forwardbody1">
<div style=3D"font-size: 10pt; font-family: Arial,Helvetica,sans-serif;">
<p>Hola desde EducaMadrid</p>
</div>
</body></html>

--=_c927eca8e881b2dbd59297847775eb5c--
```

cabeceras-1.eml.txt

```
Delivered-To: pablo@gonzaleztroyano.es
Received: by 2002:a05:6838:f30f:0:0:0:0 with SMTP id z15csp2748475nki;
        Tue, 22 Mar 2022 12:10:22 -0700 (PDT)
X-Google-Smtp-Source: ABdhPJy3+wOlfg6MS0PsN9sD0fMzvQgfTWyT4wKxmjGY2UScJHrluXd+S0iVB/YzWStH/CRv8YYR
X-Received: by 2002:a05:6000:144a:b0:203:8688:35d with SMTP id v10-20020a056000144a00b002038688035dmr23315110wrx.399.1647976222749;
        Tue, 22 Mar 2022 12:10:22 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1647976222; cv=none;
        d=google.com; s=arc-20160816;
        b=j0gGDypbVDfkm0vXw3WpWITbHUUBHl3bUQ2wjqV7N0PU6F4BSxg1C6oHayM3TjzSYE
         /lCD03MXHT6sLtl1sKd/bhzTGGszy4L084p7/ifohb4pg7PKiwDjKbq/ZEVD/+TV5hsB
         5PEfpnc/lVqakgvfaWXnOoSlJxeHLIrv73C4RopTwaGCHXFlFn/A6JmvMzvmYtiBEag+
         D5eGZYlV4zC+6wYLnOIhW+xZm5lJE9ZoaSpUEo/WAfOVTuqJ70nYetXVX+Bh2EjeI0Zx
         gXH0A+oxv5RaeAruYNxeGYjcKofxW1+mBSl5yHwaSnp884ZAKDODxgjZlnLH6QTophOt
```

```
         +GYw==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;
        h=content-language:content-transfer-encoding:mime-version:user-agent
         :date:message-id:subject:from:to:dkim-signature;
        bh=DMswiTobi3CTIswkfkqpw8IocrgObMlAL8GObemSRCk=;
        b=gJYLEn89ySDfHYvkdNDI7eYBPBJ2jvLkg+fBk72jkvoAQLzUX/HUZMjpQQfnY9lrnK
         96TF+L5D6zYrcuNMkWgs49ZZWSCkQ+tenQlfujVm3hISjBqvBNaJmYVmQnbfJJ6Fp4NQ
         3lbz6q47ocu8GPVrV1vRvqquZwu8gDTU8MS16bJpmiCFOYvDKNh0TT1fOcfYU78JU0Oz
         O4SqG/G2Nk+C/M16SqyRwLOIYF23TpVi2918471DRaqwkXecFzipLROyYKBR8FcDpREV
         ZjbovUgFbW+iw2sO0lToBuyh7YF54rDl8a2KddCJOsUOEJw80jn25uqDpzdvQDTywdQI
         OgUA==
ARC-Authentication-Results: i=1; mx.google.com;
       dkim=pass header.i=@carpet4you.site header.s=glezcloud header.b=0nhJoLiB;
       spf=pass (google.com: domain of pablo@carpet4you.site designates 93.189.91.9 as permitted sender)
smtp.mailfrom=pablo@carpet4you.site
Return-Path: <pablo@carpet4you.site>
Received: from mail.glez.cloud (mail.glez.cloud. [93.189.91.9])
        by mx.google.com with ESMTPS id 9-20020a05600c028900b0038c77be9cfbsi2812221wmk.195.2022.03.22.12.10.22
        for <pablo@gonzaleztroyano.es>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Tue, 22 Mar 2022 12:10:22 -0700 (PDT)
Received-SPF: pass (google.com: domain of pablo@carpet4you.site designates 93.189.91.9 as permitted sender) client-ip=93.189.91.9;
Authentication-Results: mx.google.com;
       dkim=pass header.i=@carpet4you.site header.s=glezcloud header.b=0nhJoLiB;
       spf=pass (google.com: domain of pablo@carpet4you.site designates 93.189.91.9 as permitted sender)
smtp.mailfrom=pablo@carpet4you.site
Received: from [127.0.0.1] (localhost [127.0.0.1]) by localhost (Mailerdaemon) with ESMTPSA id 971126EDDF for
<pablo@gonzaleztroyano.es>; Tue, 22 Mar 2022 20:10:21 +0100 (CET)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=carpet4you.site; s=glezcloud; t=1647976221;
h=from:subject:date:message-id:to:mime-version:content-type:
           content-transfer-encoding:content-language; bh=DMswiTobi3CTIswkfkqpw8IocrgObMlAL8GObemSRCk=;
b=0nhJoLiBvQvX27Hb7PRoD8oatksr90fzbhSBqcM7XeGJCxRbqwvW00z1ZYSajS4SorAmLV
4RjuSPxSvJdAX50FDbDe0lkpRJzs3PBqVIkA51bTHA8ij5hWDn1b8fXfnqFRVXM9oK3A8x n7NPcd3mXW5eUl0f//I0X1RA1zUH8qw=
To: pablo@gonzaleztroyano.es
From: "Pablo González" <pablo@carpet4you.site>
Subject: Prueba de envío desde Thunderbird
Message-ID: <3a043d9b-d083-071f-23d0-5fff41a5b0f4@carpet4you.site>
Date: Tue, 22 Mar 2022 20:10:23 +0100
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Thunderbird/78.14.0
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8; format=flowed
Content-Transfer-Encoding: 8bit
Content-Language: en-US
X-Last-TLS-Session-Version: TLSv1.3

¡Hola!

Este correo electrónico se envía desde Thunderbird.

Un saludo,

Pablo González
```

cabeceras-4.eml.txt

```
Delivered-To: [REDACTED]^^@^^gonzaleztroyano.es
Received: by 2002:a05:6838:f30f:0:0:0:0 with SMTP id z15csp4728466nki;
        Thu, 24 Mar 2022 13:26:55 -0700 (PDT)
X-Google-Smtp-Source: ABdhPJyFroezqSJ7A8cqUIrYZv9kxe4r/DuulJKTJ34KtvE0m5kRJeuhbIhTt5c3CaGwxcfMdws1
X-Received: by 2002:a05:6000:188d:b0:205:1242:485a with SMTP id a13-20020a056000188d00b002051242485amr6288513wri.495.1648153615751;
        Thu, 24 Mar 2022 13:26:55 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1648153615; cv=none;
        d=google.com; s=arc-20160816;
        b=VJHghUe4ngr3cKsip+YzDYv3JbTpacSwRsm+nlrP+BtQbzqy8xddCa5WUFFY4888Yi
         9TqPf8FuKZMkR/gUogdez+4WtOoii/bKTvHkq0XJ5X6gJMPF+vwIimjXJE4b16E2aPeH
         IXULYPWnZBI6fSGe0FJBjO2d93JjKNb4GuVBW1WYZWAL5IkQoCr0wo+IxGET35yB4YU1
         HP/7e7jkHj9ZRV2LFvSOwudN9rSW+WLjQcfF9bLU92TRvrNa6HALz7CXaWWVETemvfsB
         rtCF6Mm15AFumsmQJ+9jqpxLQ1/gf1KS4nnHS28HIoC3OQrkKM5q3415q9S3Ot04fMcI
         OXww==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;
        h=message-id:subject:date:mime-version:user-agent:to:from
         :dkim-signature;
        bh=GniEBdD3k5yKjzXTk9Uqt22hAafoJYGpUjk3RUF4hcc=;
        b=Lv4//Wv/Y6X0Cr6NCnRx8uGeiLH3ORBdpHheuEe90GDx/EWewHj0jIjf9+DkGEgA80
         Ea7nLKz5BPTmXQijsbp//KXTq/BLlnMMzpQ/uikiW/kwf3fGYI+qCaI3ch2kWvmy+hRG
         FTeSmRhq0rEUQpUjTy/uS6FlOnJHC+gFvRtkQoOTNv8niwmnfsPXV+ceYfyH38T3zXms
         sWSCreTrkO82n9IojRD0JPqAacmbQWvE4yu9VyEtBMEn1AKVbjiYbSjxlyjIogtKt3zl
         zRLSSPImQ42qQo6vc9ALlzTp1ir/vdkSH1yc8kjUNGXmVquJHoqK8Mnu/YLQ0Kq1veGN
         fdvQ==
ARC-Authentication-Results: i=1; mx.google.com;
       dkim=pass header.i=@glez-cloud.tech header.s=glez-cloud header.b=fqHXTYv0;
       spf=neutral (google.com: 93.189.91.9 is neither permitted nor denied by best guess record for domain of
ricardo^^@^^glez-cloud.tech) smtp.mailfrom=ricardo^^@^^glez-cloud.tech
```

```
Return-Path: <ricardo^^@^^glez-cloud.tech>
Received: from mail.glez.cloud (mail.glez.cloud. [93.189.91.9])
        by mx.google.com with ESMTPS id v16-20020adf8b50000000b00203edc2437csi605243wra.586.2022.03.24.13.26.55
        for <[REDACTED]^^@^^gonzaleztroyano.es>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Thu, 24 Mar 2022 13:26:55 -0700 (PDT)
Received-SPF: neutral (google.com: 93.189.91.9 is neither permitted nor denied by best guess record for domain of
ricardo^^@^^glez-cloud.tech) client-ip=93.189.91.9;
Authentication-Results: mx.google.com;
       dkim=pass header.i=@glez-cloud.tech header.s=glez-cloud header.b=fqHXTYv0;
       spf=neutral (google.com: 93.189.91.9 is neither permitted nor denied by best guess record for domain of
ricardo^^@^^glez-cloud.tech) smtp.mailfrom=ricardo^^@^^glez-cloud.tech
Received: from [127.0.0.1] (localhost [127.0.0.1]) by localhost (Mailerdaemon) with ESMTPA id F1F1A6E07D
          for <[REDACTED]^^@^^gonzaleztroyano.es>; Thu, 24 Mar 2022 21:26:54 +0100 (CET)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=glez-cloud.tech;
          s=glez-cloud; t=1648153615;
          h=from:subject:date:message-id:to:mime-version:content-type;
          bh=GniEBdD3k5yKjzXTk9Uqt22hAafoJYGpUjk3RUF4hcc=;
          b=fqHXTYv0z8BU1Nek4tt83hdtUuJWziW/6pvfC7Pzi49YahGoseIlC6LIrucAClleEXu271
          tl0AeCVihzZ2JiowClekPP6qQyAYO/zDjgNljgCUmj5Ea9hhP5DtuZHxdWY1nQnbqck56V
          OgF985JxBPOoxd2Ci86og24o74htRFA=
From: =?utf-8?q?Ricardo_Felipe_Jos=C3=A9_Gonz=C3=A1lez_Fern=C3=A1ndez?= <ricardo^^@^^glez-cloud.tech>
To: [REDACTED]^^@^^gonzaleztroyano.es
User-Agent: SOGoMail 5.5.0
MIME-Version: 1.0
Date: Thu, 24 Mar 2022 21:26:54 +0100
Subject: Prueba de =?utf-8?q?env=C3=ADo?= desde la cuenta de Ricardo
Message-ID: <44-623cd400-7-5c0e2f80@125697113>
Content-Type: multipart/alternative; boundary="----=_=-_OpenGroupware_org_NGMime-68-1648153614.552799-2------"
X-Last-TLS-Session-Version: None

------=_=-_OpenGroupware_org_NGMime-68-1648153614.552799-2------
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 12


=C2=A1Hola!

------=_=-_OpenGroupware_org_NGMime-68-1648153614.552799-2------
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 31

<html><p>=C2=A1Hola!</p></html>

------=_=-_OpenGroupware_org_NGMime-68-1648153614.552799-2--------
```

7.6.3. Archivo docker-compose de mailcow

```
version: '2.1'
services:

    unbound-mailcow:
      image: mailcow/unbound:1.15
      environment:
        - TZ=${TZ}
      volumes:
        - ./data/hooks/unbound:/hooks:Z
        - ./data/conf/unbound/unbound.conf:/etc/unbound/unbound.conf:ro,Z
      restart: always
      tty: true
      networks:
        mailcow-network:
          ipv4_address: ${IPV4_NETWORK:-172.22.1}.254
          aliases:
            - unbound

    mysql-mailcow:
      image: mariadb:10.5
      depends_on:
```

```
      - unbound-mailcow
    stop_grace_period: 45s
    volumes:
      - mysql-vol-1:/var/lib/mysql/:Z
      - mysql-socket-vol-1:/var/run/mysqld/:z
      - ./data/conf/mysql/:/etc/mysql/conf.d/:ro,Z
    environment:
      - TZ=${TZ}
      - MYSQL_ROOT_PASSWORD=${DBROOT}
      - MYSQL_DATABASE=${DBNAME}
      - MYSQL_USER=${DBUSER}
      - MYSQL_PASSWORD=${DBPASS}
      - MYSQL_INITDB_SKIP_TZINFO=1
    restart: always
    ports:
      - "${SQL_PORT:-127.0.0.1:13306}:3306"
    networks:
      mailcow-network:
        aliases:
          - mysql

  redis-mailcow:
    image: redis:6-alpine
    volumes:
      - redis-vol-1:/data/:Z
    restart: always
    ports:
      - "${REDIS_PORT:-127.0.0.1:7654}:6379"
    environment:
      - TZ=${TZ}
    sysctls:
      - net.core.somaxconn=4096
    networks:
      mailcow-network:
        ipv4_address: ${IPV4_NETWORK:-172.22.1}.249
        aliases:
          - redis

  clamd-mailcow:
    image: mailcow/clamd:1.44
    restart: always
    dns:
      - ${IPV4_NETWORK:-172.22.1}.254
    environment:
      - TZ=${TZ}
      - SKIP_CLAMD=${SKIP_CLAMD:-n}
    volumes:
      - ./data/conf/clamav/:/etc/clamav/:Z
    networks:
      mailcow-network:
        aliases:
          - clamd

  rspamd-mailcow:
    image: mailcow/rspamd:1.80
    stop_grace_period: 30s
    depends_on:
      - dovecot-mailcow
    environment:
      - TZ=${TZ}
      - IPV4_NETWORK=${IPV4_NETWORK:-172.22.1}
      - IPV6_NETWORK=${IPV6_NETWORK:-fd4d:6169:6c63:6f77::/64}
      - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
      - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
    volumes:
```

```
        - ./data/hooks/rspamd:/hooks:Z
        - ./data/conf/rspamd/custom/:/etc/rspamd/custom:z
        - ./data/conf/rspamd/override.d/:/etc/rspamd/override.d:Z
        - ./data/conf/rspamd/local.d/:/etc/rspamd/local.d:Z
        - ./data/conf/rspamd/plugins.d/:/etc/rspamd/plugins.d:Z
        - ./data/conf/rspamd/lua/:/etc/rspamd/lua/:ro,Z
        - ./data/conf/rspamd/rspamd.conf.local:/etc/rspamd/rspamd.conf.local:Z
        - ./data/conf/rspamd/rspamd.conf.override:/etc/rspamd/rspamd.conf.override:Z
        - rspamd-vol-1:/var/lib/rspamd:z
      restart: always
      hostname: rspamd
      dns:
        - ${IPV4_NETWORK:-172.22.1}.254
      networks:
        mailcow-network:
          aliases:
            - rspamd

    php-fpm-mailcow:
      image: mailcow/phpfpm:1.78
      command: "php-fpm -d date.timezone=${TZ} -d expose_php=0"
      depends_on:
        - redis-mailcow
      volumes:
        - ./data/hooks/phpfpm:/hooks:Z
        - ./data/web:/web:z
        - ./data/conf/rspamd/dynmaps:/dynmaps:ro,z
        - ./data/conf/rspamd/custom/:/rspamd_custom_maps:z
        - rspamd-vol-1:/var/lib/rspamd:z
        - mysql-socket-vol-1:/var/run/mysqld/:z
        - ./data/conf/sogo/:/etc/sogo/:z
        - ./data/conf/rspamd/meta_exporter:/meta_exporter:ro,z
        - ./data/conf/phpfpm/sogo-sso/:/etc/sogo-sso/:z
        -
./data/conf/phpfpm/php-fpm.d/pools.conf:/usr/local/etc/php-fpm.d/z-pools.conf:Z
        -
./data/conf/phpfpm/php-conf.d/opcache-recommended.ini:/usr/local/etc/php/conf.d/opcache-
recommended.ini:Z
        -
./data/conf/phpfpm/php-conf.d/upload.ini:/usr/local/etc/php/conf.d/upload.ini:Z
        -
./data/conf/phpfpm/php-conf.d/other.ini:/usr/local/etc/php/conf.d/zzz-other.ini:Z
        - ./data/conf/dovecot/global_sieve_before:/global_sieve/before:z
        - ./data/conf/dovecot/global_sieve_after:/global_sieve/after:z
        - ./data/assets/templates:/tpls:z
        - ./data/conf/nginx/:/etc/nginx/conf.d/:z
      dns:
        - ${IPV4_NETWORK:-172.22.1}.254
      environment:
        - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
        - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
        - LOG_LINES=${LOG_LINES:-9999}
        - TZ=${TZ}
        - DBNAME=${DBNAME}
        - DBUSER=${DBUSER}
        - DBPASS=${DBPASS}
        - MAILCOW_HOSTNAME=${MAILCOW_HOSTNAME}
        - MAILCOW_PASS_SCHEME=${MAILCOW_PASS_SCHEME:-BLF-CRYPT}
        - IMAP_PORT=${IMAP_PORT:-143}
        - IMAPS_PORT=${IMAPS_PORT:-993}
        - POP_PORT=${POP_PORT:-110}
        - POPS_PORT=${POPS_PORT:-995}
        - SIEVE_PORT=${SIEVE_PORT:-4190}
        - IPV4_NETWORK=${IPV4_NETWORK:-172.22.1}
        - IPV6_NETWORK=${IPV6_NETWORK:-fd4d:6169:6c63:6f77::/64}
```

```
        - SUBMISSION_PORT=${SUBMISSION_PORT:-587}
        - SMTPS_PORT=${SMTPS_PORT:-465}
        - SMTP_PORT=${SMTP_PORT:-25}
        - API_KEY=${API_KEY:-invalid}
        - API_KEY_READ_ONLY=${API_KEY_READ_ONLY:-invalid}
        - API_ALLOW_FROM=${API_ALLOW_FROM:-invalid}
        - COMPOSE_PROJECT_NAME=${COMPOSE_PROJECT_NAME:-mailcow-dockerized}
        - SKIP_SOLR=${SKIP_SOLR:-y}
        - SKIP_CLAMD=${SKIP_CLAMD:-n}
        - SKIP_SOGO=${SKIP_SOGO:-n}
        - ALLOW_ADMIN_EMAIL_LOGIN=${ALLOW_ADMIN_EMAIL_LOGIN:-n}
        - MASTER=${MASTER:-y}
        - DEV_MODE=${DEV_MODE:-n}
        - WEBAUTHN_ONLY_TRUSTED_VENDORS=${WEBAUTHN_ONLY_TRUSTED_VENDORS:-n}
      restart: always
      networks:
        mailcow-network:
          aliases:
            - phpfpm

    sogo-mailcow:
      image: mailcow/sogo:1.106
      environment:
        - DBNAME=${DBNAME}
        - DBUSER=${DBUSER}
        - DBPASS=${DBPASS}
        - TZ=${TZ}
        - LOG_LINES=${LOG_LINES:-9999}
        - MAILCOW_HOSTNAME=${MAILCOW_HOSTNAME}
        - MAILCOW_PASS_SCHEME=${MAILCOW_PASS_SCHEME:-BLF-CRYPT}
        - ACL_ANYONE=${ACL_ANYONE:-disallow}
        - ALLOW_ADMIN_EMAIL_LOGIN=${ALLOW_ADMIN_EMAIL_LOGIN:-n}
        - IPV4_NETWORK=${IPV4_NETWORK:-172.22.1}
        - SOGO_EXPIRE_SESSION=${SOGO_EXPIRE_SESSION:-480}
        - SKIP_SOGO=${SKIP_SOGO:-n}
        - MASTER=${MASTER:-y}
        - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
        - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
      dns:
        - ${IPV4_NETWORK:-172.22.1}.254
      volumes:
        - ./data/hooks/sogo:/hooks:Z
        - ./data/conf/sogo/:/etc/sogo/:z
        - ./data/web/inc/init_db.inc.php:/init_db.inc.php:Z
        -
./data/conf/sogo/custom-favicon.ico:/usr/lib/GNUstep/SOGo/WebServerResources/img/sogo.ic
o:z
        -
./data/conf/sogo/custom-theme.js:/usr/lib/GNUstep/SOGo/WebServerResources/js/theme.js:z
        -
./data/conf/sogo/custom-sogo.js:/usr/lib/GNUstep/SOGo/WebServerResources/js/custom-sogo.
js:z
        - mysql-socket-vol-1:/var/run/mysqld/:z
        - sogo-web-vol-1:/sogo_web:z
        - sogo-userdata-backup-vol-1:/sogo_backup:Z
      labels:
        ofelia.enabled: "true"
        ofelia.job-exec.sogo_sessions.schedule: "@every 1m"
        ofelia.job-exec.sogo_sessions.command: "/bin/bash -c \"[[ $${MASTER} == y ]] &&
/usr/local/bin/gosu sogo /usr/sbin/sogo-tool expire-sessions $${SOGO_EXPIRE_SESSION} ||
exit 0\""
        ofelia.job-exec.sogo_ealarms.schedule: "@every 1m"
        ofelia.job-exec.sogo_ealarms.command: "/bin/bash -c \"[[ $${MASTER} == y ]] &&
/usr/local/bin/gosu sogo /usr/sbin/sogo-ealarms-notify -p /etc/sogo/sieve.creds || exit
0\""
```

```
        ofelia.job-exec.sogo_eautoreply.schedule: "@every 5m"
        ofelia.job-exec.sogo_eautoreply.command: "/bin/bash -c \"[[ $${MASTER} == y ]]
&& /usr/local/bin/gosu sogo /usr/sbin/sogo-tool update-autoreply -p
/etc/sogo/sieve.creds || exit 0\""
        ofelia.job-exec.sogo_backup.schedule: "@every 24h"
        ofelia.job-exec.sogo_backup.command: "/bin/bash -c \"[[ $${MASTER} == y ]] &&
/usr/local/bin/gosu sogo /usr/sbin/sogo-tool backup /sogo_backup ALL || exit 0\""
      restart: always
      networks:
        mailcow-network:
          ipv4_address: ${IPV4_NETWORK:-172.22.1}.248
          aliases:
            - sogo

    dovecot-mailcow:
      image: mailcow/dovecot:1.161
      depends_on:
        - mysql-mailcow
      dns:
        - ${IPV4_NETWORK:-172.22.1}.254
      cap_add:
        - NET_BIND_SERVICE
      volumes:
        - ./data/hooks/dovecot:/hooks:Z
        - ./data/conf/dovecot:/etc/dovecot:z
        - ./data/assets/ssl:/etc/ssl/mail/:ro,z
        - ./data/conf/sogo/:/etc/sogo/:z
        - ./data/conf/phpfpm/sogo-sso/:/etc/phpfpm/:z
        - vmail-vol-1:/var/vmail:Z
        - vmail-index-vol-1:/var/vmail_index:Z
        - crypt-vol-1:/mail_crypt/:z
        - ./data/conf/rspamd/custom/:/etc/rspamd/custom:z
        - ./data/assets/templates:/templates:z
        - rspamd-vol-1:/var/lib/rspamd:z
        - mysql-socket-vol-1:/var/run/mysqld/:z
      environment:
        - DOVECOT_MASTER_USER=${DOVECOT_MASTER_USER:-}
        - DOVECOT_MASTER_PASS=${DOVECOT_MASTER_PASS:-}
        - LOG_LINES=${LOG_LINES:-9999}
        - DBNAME=${DBNAME}
        - DBUSER=${DBUSER}
        - DBPASS=${DBPASS}
        - TZ=${TZ}
        - MAILCOW_HOSTNAME=${MAILCOW_HOSTNAME}
        - MAILCOW_PASS_SCHEME=${MAILCOW_PASS_SCHEME:-BLF-CRYPT}
        - IPV4_NETWORK=${IPV4_NETWORK:-172.22.1}
        - ALLOW_ADMIN_EMAIL_LOGIN=${ALLOW_ADMIN_EMAIL_LOGIN:-n}
        - MAILDIR_GC_TIME=${MAILDIR_GC_TIME:-7200}
        - ACL_ANYONE=${ACL_ANYONE:-disallow}
        - SKIP_SOLR=${SKIP_SOLR:-y}
        - MAILDIR_SUB=${MAILDIR_SUB:-}
        - MASTER=${MASTER:-y}
        - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
        - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
        - COMPOSE_PROJECT_NAME=${COMPOSE_PROJECT_NAME:-mailcow-dockerized}
      ports:
        - "${DOVEADM_PORT:-127.0.0.1:19991}:12345"
        - "${IMAP_PORT:-143}:143"
        - "${IMAPS_PORT:-993}:993"
        - "${POP_PORT:-110}:110"
        - "${POPS_PORT:-995}:995"
        - "${SIEVE_PORT:-4190}:4190"
      restart: always
      tty: true
      labels:
```

```
        ofelia.enabled: "true"
        ofelia.job-exec.dovecot_imapsync_runner.schedule: "@every 1m"
        ofelia.job-exec.dovecot_imapsync_runner.no-overlap: "true"
        ofelia.job-exec.dovecot_imapsync_runner.command: "/bin/bash -c \"[[ $${MASTER}
== y ]] && /usr/local/bin/gosu nobody /usr/local/bin/imapsync_runner.pl || exit 0\""
        ofelia.job-exec.dovecot_trim_logs.schedule: "@every 1m"
        ofelia.job-exec.dovecot_trim_logs.command: "/bin/bash -c \"[[ $${MASTER} == y ]]
&& /usr/local/bin/gosu vmail /usr/local/bin/trim_logs.sh || exit 0\""
        ofelia.job-exec.dovecot_quarantine.schedule: "@every 20m"
        ofelia.job-exec.dovecot_quarantine.command: "/bin/bash -c \"[[ $${MASTER} == y
]] && /usr/local/bin/gosu vmail /usr/local/bin/quarantine_notify.py || exit 0\""
        ofelia.job-exec.dovecot_clean_q_aged.schedule: "@every 24h"
        ofelia.job-exec.dovecot_clean_q_aged.command: "/bin/bash -c \"[[ $${MASTER} == y
]] && /usr/local/bin/gosu vmail /usr/local/bin/clean_q_aged.sh || exit 0\""
        ofelia.job-exec.dovecot_maildir_gc.schedule: "@every 30m"
        ofelia.job-exec.dovecot_maildir_gc.command: "/bin/bash -c \"source
/source_env.sh ; /usr/local/bin/gosu vmail /usr/local/bin/maildir_gc.sh\""
        ofelia.job-exec.dovecot_sarules.schedule: "@every 24h"
        ofelia.job-exec.dovecot_sarules.command: "/bin/bash -c
\"/usr/local/bin/sa-rules.sh\""
        ofelia.job-exec.dovecot_fts.schedule: "@every 24h"
        ofelia.job-exec.dovecot_fts.command: "/usr/bin/curl
http://solr:8983/solr/dovecot-fts/update?optimize=true"
        ofelia.job-exec.dovecot_repl_health.schedule: "@every 5m"
        ofelia.job-exec.dovecot_repl_health.command: "/bin/bash -c \"/usr/local/bin/gosu
vmail /usr/local/bin/repl_health.sh\""
      ulimits:
        nproc: 65535
        nofile:
          soft: 20000
          hard: 40000
      networks:
        mailcow-network:
          ipv4_address: ${IPV4_NETWORK:-172.22.1}.250
          aliases:
            - dovecot

    postfix-mailcow:
      image: mailcow/postfix:1.66
      depends_on:
        - mysql-mailcow
      volumes:
        - ./data/hooks/postfix:/hooks:Z
        - ./data/conf/postfix:/opt/postfix/conf:z
        - ./data/assets/ssl:/etc/ssl/mail/:ro,z
        - postfix-vol-1:/var/spool/postfix:z
        - crypt-vol-1:/var/lib/zeyple:z
        - rspamd-vol-1:/var/lib/rspamd:z
        - mysql-socket-vol-1:/var/run/mysqld/:z
      environment:
        - LOG_LINES=${LOG_LINES:-9999}
        - TZ=${TZ}
        - DBNAME=${DBNAME}
        - DBUSER=${DBUSER}
        - DBPASS=${DBPASS}
        - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
        - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
        - MAILCOW_HOSTNAME=${MAILCOW_HOSTNAME}
      cap_add:
        - NET_BIND_SERVICE
      ports:
        - "${SMTP_PORT:-25}:25"
        - "${SMTPS_PORT:-465}:465"
        - "${SUBMISSION_PORT:-587}:587"
      restart: always
```

```
      dns:
        - ${IPV4_NETWORK:-172.22.1}.254
      networks:
        mailcow-network:
          ipv4_address: ${IPV4_NETWORK:-172.22.1}.253
          aliases:
            - postfix

    memcached-mailcow:
      image: memcached:alpine
      restart: always
      environment:
        - TZ=${TZ}
      networks:
        mailcow-network:
          aliases:
            - memcached

    nginx-mailcow:
      depends_on:
        - sogo-mailcow
        - php-fpm-mailcow
        - redis-mailcow
      image: nginx:mainline-alpine
      dns:
        - ${IPV4_NETWORK:-172.22.1}.254
      command: /bin/sh -c "envsubst < /etc/nginx/conf.d/templates/listen_plain.template
> /etc/nginx/conf.d/listen_plain.active &&
        envsubst < /etc/nginx/conf.d/templates/listen_ssl.template >
/etc/nginx/conf.d/listen_ssl.active &&
        envsubst < /etc/nginx/conf.d/templates/sogo.template >
/etc/nginx/conf.d/sogo.active &&
        . /etc/nginx/conf.d/templates/server_name.template.sh >
/etc/nginx/conf.d/server_name.active &&
        . /etc/nginx/conf.d/templates/sites.template.sh > /etc/nginx/conf.d/sites.active
&&
        . /etc/nginx/conf.d/templates/sogo_eas.template.sh >
/etc/nginx/conf.d/sogo_eas.active &&
        nginx -qt &&
        until ping phpfpm -c1 > /dev/null; do sleep 1; done &&
        until ping sogo -c1 > /dev/null; do sleep 1; done &&
        until ping redis -c1 > /dev/null; do sleep 1; done &&
        until ping rspamd -c1 > /dev/null; do sleep 1; done &&
        exec nginx -g 'daemon off;'"
      environment:
        - HTTPS_PORT=${HTTPS_PORT:-443}
        - HTTP_PORT=${HTTP_PORT:-80}
        - MAILCOW_HOSTNAME=${MAILCOW_HOSTNAME}
        - IPV4_NETWORK=${IPV4_NETWORK:-172.22.1}
        - TZ=${TZ}
        - SKIP_SOGO=${SKIP_SOGO:-n}
        - ALLOW_ADMIN_EMAIL_LOGIN=${ALLOW_ADMIN_EMAIL_LOGIN:-n}
        - ADDITIONAL_SERVER_NAMES=${ADDITIONAL_SERVER_NAMES:-}
      volumes:
        - ./data/web:/web:ro,z
        - ./data/conf/rspamd/dynmaps:/dynmaps:ro,z
        - ./data/assets/ssl/:/etc/ssl/mail/:ro,z
        - ./data/conf/nginx/:/etc/nginx/conf.d/:z
        - ./data/conf/rspamd/meta_exporter:/meta_exporter:ro,z
        - sogo-web-vol-1:/usr/lib/GNUstep/SOGo/:z
      ports:
        - "${HTTPS_BIND:-:}:${HTTPS_PORT:-443}:${HTTPS_PORT:-443}"
        - "${HTTP_BIND:-:}:${HTTP_PORT:-80}:${HTTP_PORT:-80}"
      restart: always
      networks:
```

```
      mailcow-network:
        aliases:
          - nginx

  acme-mailcow:
    depends_on:
      - nginx-mailcow
    image: mailcow/acme:1.81
    dns:
      - ${IPV4_NETWORK:-172.22.1}.254
    environment:
      - LOG_LINES=${LOG_LINES:-9999}
      - ACME_CONTACT=${ACME_CONTACT:-}
      - ADDITIONAL_SAN=${ADDITIONAL_SAN}
      - MAILCOW_HOSTNAME=${MAILCOW_HOSTNAME}
      - DBNAME=${DBNAME}
      - DBUSER=${DBUSER}
      - DBPASS=${DBPASS}
      - SKIP_LETS_ENCRYPT=${SKIP_LETS_ENCRYPT:-n}
      - COMPOSE_PROJECT_NAME=${COMPOSE_PROJECT_NAME:-mailcow-dockerized}
      - DIRECTORY_URL=${DIRECTORY_URL:-}
      - ENABLE_SSL_SNI=${ENABLE_SSL_SNI:-n}
      - SKIP_IP_CHECK=${SKIP_IP_CHECK:-n}
      - SKIP_HTTP_VERIFICATION=${SKIP_HTTP_VERIFICATION:-n}
      - ONLY_MAILCOW_HOSTNAME=${ONLY_MAILCOW_HOSTNAME:-n}
      - LE_STAGING=${LE_STAGING:-n}
      - TZ=${TZ}
      - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
      - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
      - SNAT_TO_SOURCE=${SNAT_TO_SOURCE:-n}
      - SNAT6_TO_SOURCE=${SNAT6_TO_SOURCE:-n}
    volumes:
      - ./data/web/.well-known/acme-challenge:/var/www/acme:z
      - ./data/assets/ssl:/var/lib/acme/:z
      - ./data/assets/ssl-example:/var/lib/ssl-example/:ro,Z
      - mysql-socket-vol-1:/var/run/mysqld/:z
    restart: always
    networks:
      mailcow-network:
        aliases:
          - acme

  netfilter-mailcow:
    image: mailcow/netfilter:1.46
    stop_grace_period: 30s
    depends_on:
      - dovecot-mailcow
      - postfix-mailcow
      - sogo-mailcow
      - php-fpm-mailcow
      - redis-mailcow
    restart: always
    privileged: true
    environment:
      - TZ=${TZ}
      - IPV4_NETWORK=${IPV4_NETWORK:-172.22.1}
      - IPV6_NETWORK=${IPV6_NETWORK:-fd4d:6169:6c63:6f77::/64}
      - SNAT_TO_SOURCE=${SNAT_TO_SOURCE:-n}
      - SNAT6_TO_SOURCE=${SNAT6_TO_SOURCE:-n}
      - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
      - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
    network_mode: "host"
    volumes:
      - /lib/modules:/lib/modules:ro
```

```
  watchdog-mailcow:
    image: mailcow/watchdog:1.96
    dns:
      - ${IPV4_NETWORK:-172.22.1}.254
    tmpfs:
      - /tmp
    volumes:
      - rspamd-vol-1:/var/lib/rspamd:z
      - mysql-socket-vol-1:/var/run/mysqld/:z
      - postfix-vol-1:/var/spool/postfix:z
      - ./data/assets/ssl:/etc/ssl/mail/:ro,z
    restart: always
    environment:
      - IPV6_NETWORK=${IPV6_NETWORK:-fd4d:6169:6c63:6f77::/64}
      - LOG_LINES=${LOG_LINES:-9999}
      - TZ=${TZ}
      - DBNAME=${DBNAME}
      - DBUSER=${DBUSER}
      - DBPASS=${DBPASS}
      - DBROOT=${DBROOT}
      - USE_WATCHDOG=${USE_WATCHDOG:-n}
      - WATCHDOG_NOTIFY_EMAIL=${WATCHDOG_NOTIFY_EMAIL:-}
      - WATCHDOG_NOTIFY_BAN=${WATCHDOG_NOTIFY_BAN:-y}
      - WATCHDOG_SUBJECT=${WATCHDOG_SUBJECT:-Watchdog ALERT}
      - WATCHDOG_EXTERNAL_CHECKS=${WATCHDOG_EXTERNAL_CHECKS:-n}
      - WATCHDOG_MYSQL_REPLICATION_CHECKS=${WATCHDOG_MYSQL_REPLICATION_CHECKS:-n}
      - WATCHDOG_VERBOSE=${WATCHDOG_VERBOSE:-n}
      - MAILCOW_HOSTNAME=${MAILCOW_HOSTNAME}
      - COMPOSE_PROJECT_NAME=${COMPOSE_PROJECT_NAME:-mailcow-dockerized}
      - IPV4_NETWORK=${IPV4_NETWORK:-172.22.1}
      - IP_BY_DOCKER_API=${IP_BY_DOCKER_API:-0}
      - CHECK_UNBOUND=${CHECK_UNBOUND:-1}
      - SKIP_CLAMD=${SKIP_CLAMD:-n}
      - SKIP_LETS_ENCRYPT=${SKIP_LETS_ENCRYPT:-n}
      - SKIP_SOGO=${SKIP_SOGO:-n}
      - HTTPS_PORT=${HTTPS_PORT:-443}
      - REDIS_SLAVEOF_IP=${REDIS_SLAVEOF_IP:-}
      - REDIS_SLAVEOF_PORT=${REDIS_SLAVEOF_PORT:-}
      - EXTERNAL_CHECKS_THRESHOLD=${EXTERNAL_CHECKS_THRESHOLD:-1}
      - NGINX_THRESHOLD=${NGINX_THRESHOLD:-5}
      - UNBOUND_THRESHOLD=${UNBOUND_THRESHOLD:-5}
      - REDIS_THRESHOLD=${REDIS_THRESHOLD:-5}
      - MYSQL_THRESHOLD=${MYSQL_THRESHOLD:-5}
      - MYSQL_REPLICATION_THRESHOLD=${MYSQL_REPLICATION_THRESHOLD:-1}
      - SOGO_THRESHOLD=${SOGO_THRESHOLD:-3}
      - POSTFIX_THRESHOLD=${POSTFIX_THRESHOLD:-8}
      - CLAMD_THRESHOLD=${CLAMD_THRESHOLD:-15}
      - DOVECOT_THRESHOLD=${DOVECOT_THRESHOLD:-12}
      - DOVECOT_REPL_THRESHOLD=${DOVECOT_REPL_THRESHOLD:-20}
      - PHPFPM_THRESHOLD=${PHPFPM_THRESHOLD:-5}
      - RATELIMIT_THRESHOLD=${RATELIMIT_THRESHOLD:-1}
      - FAIL2BAN_THRESHOLD=${FAIL2BAN_THRESHOLD:-1}
      - ACME_THRESHOLD=${ACME_THRESHOLD:-1}
      - RSPAMD_THRESHOLD=${RSPAMD_THRESHOLD:-5}
      - OLEFY_THRESHOLD=${OLEFY_THRESHOLD:-5}
      - MAILQ_THRESHOLD=${MAILQ_THRESHOLD:-20}
      - MAILQ_CRIT=${MAILQ_CRIT:-30}
    networks:
      mailcow-network:
        aliases:
          - watchdog

  dockerapi-mailcow:
    image: mailcow/dockerapi:1.41
    security_opt:
```

```
      - label=disable
    restart: always
    oom_kill_disable: true
    dns:
      - ${IPV4_NETWORK:-172.22.1}.254
    environment:
      - DBROOT=${DBROOT}
      - TZ=${TZ}
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
    networks:
      mailcow-network:
        aliases:
          - dockerapi

  solr-mailcow:
    image: mailcow/solr:1.8.1
    restart: always
    volumes:
      - solr-vol-1:/opt/solr/server/solr/dovecot-fts/data:Z
    ports:
      - "${SOLR_PORT:-127.0.0.1:18983}:8983"
    environment:
      - TZ=${TZ}
      - SOLR_HEAP=${SOLR_HEAP:-1024}
      - SKIP_SOLR=${SKIP_SOLR:-y}
    networks:
      mailcow-network:
        aliases:
          - solr

  olefy-mailcow:
    image: mailcow/olefy:1.9
    restart: always
    environment:
      - TZ=${TZ}
      - OLEFY_BINDADDRESS=0.0.0.0
      - OLEFY_BINDPORT=10055
      - OLEFY_TMPDIR=/tmp
      - OLEFY_PYTHON_PATH=/usr/bin/python3
      - OLEFY_OLEVBA_PATH=/usr/bin/olevba
      - OLEFY_LOGLVL=20
      - OLEFY_MINLENGTH=500
      - OLEFY_DEL_TMP=1
    networks:
      mailcow-network:
        aliases:
          - olefy

  ofelia-mailcow:
    image: mcuadros/ofelia:latest
    restart: always
    command: daemon --docker
    environment:
      - TZ=${TZ}
    depends_on:
      - sogo-mailcow
      - dovecot-mailcow
    labels:
      ofelia.enabled: "true"
    security_opt:
      - label=disable
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
    networks:
```

```
        mailcow-network:
          aliases:
            - ofelia

    ipv6nat-mailcow:
      depends_on:
        - unbound-mailcow
        - mysql-mailcow
        - redis-mailcow
        - clamd-mailcow
        - rspamd-mailcow
        - php-fpm-mailcow
        - sogo-mailcow
        - dovecot-mailcow
        - postfix-mailcow
        - memcached-mailcow
        - nginx-mailcow
        - acme-mailcow
        - netfilter-mailcow
        - watchdog-mailcow
        - dockerapi-mailcow
        - solr-mailcow
      environment:
        - TZ=${TZ}
      image: robbertkl/ipv6nat
      security_opt:
        - label=disable
      restart: always
      privileged: true
      network_mode: "host"
      volumes:
        - /var/run/docker.sock:/var/run/docker.sock:ro
        - /lib/modules:/lib/modules:ro

networks:
  mailcow-network:
    driver: bridge
    driver_opts:
      com.docker.network.bridge.name: br-mailcow
    enable_ipv6: true
    ipam:
      driver: default
      config:
        - subnet: ${IPV4_NETWORK:-172.22.1}.0/24
        - subnet: ${IPV6_NETWORK:-fd4d:6169:6c63:6f77::/64}

volumes:
  vmail-vol-1:
  vmail-index-vol-1:
  mysql-vol-1:
  mysql-socket-vol-1:
  redis-vol-1:
  rspamd-vol-1:
  solr-vol-1:
  postfix-vol-1:
  crypt-vol-1:
  sogo-web-vol-1:
  sogo-userdata-backup-vol-1:
```

7.7. Anexo VII: Códigos relativos al servicio web

gestion.sh

```
#!/bin/bash

source source/borrar.sh
source source/config_wp.sh
source source/crear_apache.sh
source source/crear_usuario.sh
source source/crear_wp.sh
source source/envio_email.sh
source source/menu.sh
source source/modificar.sh
source source/listar.sh
source source/conf_inicial.sh
source source/secrets.sh
source source/cf_updater.sh
source source/borrar_hard.sh
source source/save_passwd.sh
source source/app_list.sh
source source/show_header.sh
source source/add_app.sh
source source/cert_creation.sh
source source/install_prestashop.sh

if [ "${EUID}" -ne 0 ]
  then echo "Este script de gestión solo puede ser ejecutado por el
usuario root"
  exit
fi

menu
```

add_app.sh

```
function add_app() {
    show_header

    echo -e "Usuarios del sistema web: \n "
        cat /etc/passwd | grep '/var/www' | cut -d ':' -f 1
        echo -e "\n -- FIN DE LA LISTA -- \n \n"

    read -p "Indicar el usuario sobre el que se desea listar las
aplicaciones instaladas: " usuario_a_listar_apps

    check_usuario_existe=$(cat /etc/passwd | grep "/var/www" | cut -d
```

```
":" -f 1 | grep -w ${usuario_a_listar_apps})
    if [[ ${check_usuario_existe} != ${usuario_a_listar_apps} ]]; then

        echo "El usuario indicado no existe"
        read "Volver al menú..."
        menu
    else
        app_list tabla ${usuario_a_listar_apps}
        apps_instaladas=${bin_apps} #40

    fi

    echo -e "\n ¿Qué aplicación desea instalar?\n     1. Instalar
WordPress\n     2. Instalar PrestaShop"
    read -p " Indique aplicación a instalar [1/2]: " app_a_instalar

    if [[ ${apps_instaladas:1:1} = 1 ]] && [[ ${app_a_instalar} = 1 ]];
then
        echo -e "\n\033[1mERROR:\033[0m WordPress ya está instalada para
el usuario ${usuario_a_listar_apps}"
        read -p "Pulse cualquier tecla para volver al menú..." trash
        menu
    fi

    if [[ ${apps_instaladas:0:1} = 1 ]] && [[ ${app_a_instalar} = 2 ]];
then
        echo -e "\n\033[1mERROR:\033[0m PrestaShop ya está instalada
para el usuario ${usuario_a_listar_apps}"
        read -p "Pulse cualquier tecla para volver al menú..." trash
        menu
    fi

    if [[ ${app_a_instalar} = 1 ]]; then
        password_generada=$(openssl rand -base64 12)
        crear_wp ${usuario_a_listar_apps} ${password_generada}
        config_wp ${usuario_a_listar_apps} ${password_generada}

        cf_updater ${usuario_a_listar_apps} blog

        cert_creation "blog.${usuario_a_listar_apps}"

        destination="/root/app_list/${usuario_a_listar_apps}"
        if [[ ${apps_instaladas:1:1} = 1 ]]; then # PS ya está instalado
            echo "111" > ${destination}
        else # WP no instalado
            echo "011" > ${destination}
        fi

        read -p "Indique el correo electrónico del cliente: "
correo_cliente
```

```
        mail_regex="^[a-zA-Z0-9_-]+@[a-zA-Z_]+?\.[a-zA-Z]{2,12}$"
        until [[ ${correo_cliente} =~ ${mail_regex} ]];
        do
            echo -e "\e[5mERROR\e[0m: correo no válido.\n"
            read -p "Indique el correo electrónico del cliente: "
correo_cliente
        done
        envio_email ${usuario_nuevo} not_applicable ${correo_cliente} 3
blog

        echo "Se ha instalado correctamente la aplicación WordPress para
el usuario ${usuario_a_listar_apps}"
        read -p "Pulse cualquier tecla para continuar" caca
        menu
    fi

    if [[ ${app_a_instalar} = 2 ]]; then
        install_prestashop ${usuario_a_listar_apps}

        destination="/root/app_list/${usuario_a_listar_apps}"
        if [[ ${apps_instaladas:1:1} = 1 ]]; then # WP ya está instalado
            echo "111" > ${destination}
        else # WP no instalado
            echo "101" > ${destination}
        fi
    fi

    if [[ ${app_a_instalar} != 1 ]] || [[ ${app_a_instalar} != 1 ]];
then
        echo -e "\n\033[1mERROR:\033[0m Opción no válida"
        read -p "Pulse cualquier tecla para volver al menú..." trash
        menu
    fi


}
```

app_list.sh

```
function app_list() {
    # Modo "silent":
        # No muestra header
        # Pide usuario, comprueba
        # Retorna valor binario
    # Modo "bonito"
        # Muestra header
        # Pide usuario, comprueba
```

```
        # Muestra tablita
    # Modo "tabla"
        # No muestra header
        # Pide usuario, comprueba
        # Muestra tablita.
        # Retorna valor binario
    # VARS:
        # $1: mode: "silent" or "bonito" or "tabla"
        # $2: user: id usuario ya comprobado

    if [[ ${1} = "bonito" ]]; then
        show_header
    fi

    if [[ ${#} = 1 ]]; then
        # Listar usuarios
            echo -e "Usuarios del sistema web: \n "
                cat /etc/passwd | grep '/var/www' | cut -d ':' -f 1
                echo -e "\n -- FIN DE LA LISTA -- \n \n"

        # Pedir usuario a modificar
            read -p "Indicar el usuario sobre el que se desea listar las
aplicaciones instaladas: " usuario_a_listar_apps

        # Comprobar si el usuario existe
    fi
    if [[ ${#} = 2 ]]; then
        usuario_a_listar_apps=${2}
    fi
        check_usuario_existe=$(cat /etc/passwd | grep "/var/www" | cut -d ":" -f
1 | grep -w ${usuario_a_listar_apps})
        if [[ ${check_usuario_existe} != ${usuario_a_listar_apps} ]]; then

            echo "El usuario indicado no existe"
            read "Volver al menú..."
            menu
        else
        # Si existe, actuar:
            bin_apps=$(cat /root/app_list/${usuario_a_listar_apps})

            if [[ ${1} = "silent" ]]; then
                return "${bin_apps}"
            fi

            if [[ ${bin_apps:0:1} = 0 ]]; then
                has_ps="❌"
                elif [[ ${bin_apps:0:1} = 1 ]]; then
                    has_ps="✅"
                else
                    has_ps="?"
```

```
            fi

            if [[ ${bin_apps:1:1} = 0 ]]; then
                has_wp="❌"
                elif [[ ${bin_apps:1:1} = 1 ]]; then
                    has_wp="✅"
                else
                    has_wp="?"
            fi

            if [[ ${bin_apps:2:1} = 0 ]]; then
                has_ss="❌"
                elif [[ ${bin_apps:2:1} = 1 ]]; then
                    has_ss="✅"
                else
                    has_ss="?"
            fi

            if [[ ${1} = "bonito" ]] || [[ ${1} = "tabla" ]]; then

                show_header
                echo -e "   Para el usuario:    ${usuario_a_listar_apps}\n"
                echo "|================================================|"
                echo "|                          ||                    |"
                echo "|     Sitio estático       ||          ${has_ss}         |"
                echo "|                          ||                    |"
                echo "|================================================|"
                echo "|                          ||                    |"
                echo "|     Sitio WordPress      ||          ${has_wp}         |"
                echo "|                          ||                    |"
                echo "|================================================|"
                echo "|                          ||                    |"
                echo "|    Sitio PrestaShop      ||          ${has_ps}         |"
                echo "|                          ||                    |"
                echo "|================================================|"
                echo ""
                if [[ ${1} = "tabla" ]]; then
                    export bin_apps
                fi
            echo -e "\n \n Volver al menú..."
            read
            menu
            fi
        fi
}
```

borrar.sh

```
# Notas del fichero
        # VARS: No incoming data
        # MUST DO:
            # Disable user Linux
                # AT +30d to remove user
            # Disable apache & WP site
                # Remove access WP
                # AT +30d Delete DB & site data
function borrar (){

    # Listar usuarios
        echo -e "Usuarios del sistema web: \n "
            grep '/var/www'  < /etc/passwd | cut -d ':' -f 1
            echo -e "\n -- FIN DE LA LISTA -- \n \n"

    # Pedir usuario a modificar
        read -p "¿Qué usuario desea borrar? " usuario_a_borrar

    # Comprobar si el usuario existe
        check_usuario_existe=$(cat /etc/passwd | grep "/var/www" | cut
-d ":" -f 1 | grep -w ${usuario_a_borrar})

        if [[ ${check_usuario_existe} != "${usuario_a_borrar}" ]]; then

    # Si el usuario NO existe, error y volver
            echo "El usuario indicado no existe"
            read -p "Pulse cualquier tecla para volver al menú inicial "
caca
            menu
        else

    # Si el usuario SÍ existe, proceder:

        # Disable user
            usermod -L ${usuario_a_borrar}

        # AT +30 remove user
            at now + 30 days "userdel -f ${usuario_a_borrar}"

        # Disable apache & WP site
            a2dissite ${usuario_a_borrar}.conf > /dev/null
            a2dissite ${usuario_a_borrar}-le-ssl.conf > /dev/null
            a2dissite wp_${usuario_a_borrar}.conf > /dev/null
            a2dissite wp_${usuario_a_borrar}-le-ssl.conf > /dev/null
            a2dissite tienda_${usuario_a_borrar}.conf > /dev/null
```

```
            a2dissite tienda_${usuario_a_borrar}-le-ssl.conf > /dev/null
            mysql -e "REVOKE ALL PRIVILEGES ON wp_${usuario_a_borrar}.*
FROM ${usuario_a_borrar};"
            mysql -e "REVOKE ALL PRIVILEGES ON
${usuario_a_borrar}_tienda.* FROM
${usuario_a_borrar}_tienda'@'localhost';"
            systemctl reload apache2
        # AT +30d Delete DB & site data
            echo "rm -Rf /var/www/${usuario_a_borrar}" | at now + 30
days
            echo "mysql -e 'DROP DATABASE IF EXISTS
wp_${usuario_a_borrar};'" | at now + 30 days
            echo "mysql -e 'DROP DATABASE IF EXISTS
${usuario_a_borrar}_tienda;'" | at now + 30 days
            echo "mysql -e 'DROP USER IF EXISTS ${usuario_a_borrar};'" |
at now + 30 days
            echo "mysql -e 'DROP USER IF EXISTS
${usuario_a_borrar}_tienda;'" | at now + 30 days
        #Confirmación
            echo "${usuario_a_borrar}, sus sitios y accesos hasn sido
deshabilitados correctamente"
            echo "${usuario_a_borrar} y sus sitios han sido programados
para eliminación en 30 días."
            read -p "Pulse intro para volver al menú" caca
        menu
    fi

}
```

borrar_hard.sh

```
function borrar_hard () {
    echo -e "Usuarios del sistema web: \n "
        grep '/var/www'  < /etc/passwd | cut -d ':' -f 1
        echo -e "\n -- FIN DE LA LISTA -- \n \n"
    read -p "¿Qué usuario desea borrar? " usuario_a_borrar
    check_usuario_existe=$(cat /etc/passwd | grep "/var/www" | cut -d
":" -f 1 | grep -w ${usuario_a_borrar})
    if [[ ${check_usuario_existe} != "${usuario_a_borrar}" ]]; then
        echo "El usuario indicado no existe"
        read -p "Pulse cualquier tecla para volver al menú inicial "
caca
        menu
    else
        userdel -f ${usuario_a_borrar}
        a2dissite ${usuario_a_borrar}.conf > /dev/null
```

```
        a2dissite ${usuario_a_borrar}-le-ssl.conf > /dev/null
        a2dissite wp_${usuario_a_borrar}.conf > /dev/null
        a2dissite wp_${usuario_a_borrar}-le-ssl.conf > /dev/null
        a2dissite tienda_${usuario_a_borrar}.conf > /dev/null
        a2dissite tienda_${usuario_a_borrar}-le-ssl.conf > /dev/null
        mysql -e "REVOKE ALL PRIVILEGES ON wp_${usuario_a_borrar}.* FROM
${usuario_a_borrar};"
        mysql -e "REVOKE ALL PRIVILEGES ON ${usuario_a_borrar}_tienda.*
FROM ${usuario_a_borrar}_tienda'@'localhost';"
        systemctl reload apache2 > /dev/null
        rm -Rf /var/www/${usuario_a_borrar}
        mysql -e "DROP DATABASE IF EXISTS wp_${usuario_a_borrar};"
        mysql -e "DROP DATABASE IF EXISTS ${usuario_a_borrar}_tienda;'"
        mysql -e "DROP USER IF EXISTS ${usuario_a_borrar}_tienda;"
        mysql -e "DROP USER IF EXISTS ${usuario_a_borrar};"
        echo "${usuario_a_borrar} y sus sitios han sido eliminados."
        read -p "Pulse intro para volver al menú"
        menu
    fi
}
```

cert_creation.sh

```
function cert_creation(){
    # VARS:
        # $1 = subdominio a crear
    subdomain=${1}
    echo -e "\n"
    read -p "¿Estamos ante un sitio de pruebas? [s/*]: "
ans_sitio_pruebas
    echo "Recibido. Generando y aplicando certificados..."
        ans_sitio_pruebas=${ans_sitio_pruebas,,}

    if [[ ${ans_sitio_pruebas} = "s" ]]; then
        certbot --apache --test-cert --quiet --redirect --agree-tos
--email support@villablanca.me -d ${subdomain}.${global_base_domain}
    else
        certbot --apache --redirect --quiet --agree-tos --email
support@villablanca.me -d ${subdomain}.${global_base_domain}
    fi
    echo "Certificados generados y aplicados correctamente."
}
```

cf_updater.sh

```
    #Notas:
        # Recibe ${usuario_nuevo} (1) desde crear_usuario.sh
        # Recibe el servicio, en caso de ser necesario ($2)
        #PEJ:
            # blog.pepe.villablanca.me --> ${2}.{$1}.villablanca.me
function cf_updater(){
    user_subdomain=$1
    service_subdomain=$2
    ip_equipo=$(curl -sS ifconfig.me)

    if [[ $# = 1 ]]; then
        curl --silent -X POST
"https://api.cloudflare.com/client/v4/zones/${global_cf_zone}/dns_record
s" \
        -H "Authorization: Bearer ${global_cf_token} " \
        -H "Content-Type: application/json" \
        --data
'{"type":"A","name":"'${user_subdomain}.${global_base_domain}'","content
":"'${ip_equipo}'","ttl":3600,"proxied":false}' | jq .success

        echo -e "Comprobando la resulución del dominio
${user_subdomain}.${global_base_domain}\n Por favor, espera..."
        sleep 5
        ip_resultado=$(dig A ${user_subdomain}.${global_base_domain}
+short @1.1.1.1)

        if [[ ${ip_resultado} = ${ip_equipo} ]]; then

            echo -e "¡Excelente! El registro creado es válido\n"
            read -p "Pulse cualquier tecla para continuar el proceso."
        else
            echo -e "No se ha podido comprobar la correcta resolución
del dominio. \n\n No nos alarmemos.\n Prueba el siguiente comando: \"dig
A ${user_subdomain}.${global_base_domain} +short\" \n\nEl resultado debe
ser: ${ip_equipo} \n\n De no resolverse, revisa en Cloudflare.\n\n"
        fi

    elif [[ $# = 2 ]]; then
        curl --silent -X POST
"https://api.cloudflare.com/client/v4/zones/${global_cf_zone}/dns_record
s" \
        -H "Authorization: Bearer ${global_cf_token} " \
        -H "Content-Type: application/json" \
        --data
'{"type":"A","name":"'${service_subdomain}.${user_subdomain}.${global_ba
```

```
se_domain}'","content":"'${ip_equipo}'","ttl":3600,"proxied":false}' |
jq .success

        echo -e "Comprobando la resulución del dominio
${service_subdomain}.${user_subdomain}.${global_base_domain}\n Por
favor, espera..."
        sleep 5
        ip_resultado=$(dig A
${service_subdomain}.${user_subdomain}.${global_base_domain} +short
@1.1.1.1)

        if [[ ${ip_resultado} = ${ip_equipo} ]]; then
            echo -e "¡Excelente! El registro creado es válido\n"
            read -p "Pulse cualquier tecla para continuar el proceso."
        else
            echo -e "No se ha podido comprobar la correcta resolución
del dominio. \n\n No nos alarmemos.\n Prueba el siguiente comando: \"dig
A ${service_subdomain}.${user_subdomain}.${global_base_domain} +short
@1.1.1.1\" \n\nEl resultado debe ser: ${ip_equipo} \n\n De no
resolverse, revisa en Cloudflare.\n\n"
        fi
    fi
}
```

conf_inicial.sh

```
conf_inicial(){
        clear
        echo ""
        echo ""
        show_header
    echo -e "Actualizando la lista de paquetes disponibles en los
repositorios..."
    apt-get update -y > /tmp/conf_inicial.log
    if [[ $? = 0 ]]; then
        echo -e "\t¡Hecho!\nInstalando paquetes necesarios..."
        else
            echo -e "¡Cachis! Parece que se ha producido un error.
Revise los logs en /tmp/conf_inicial.log"
            read -p "Pulsa cualquier tecla para continuar..."
            exit
    fi
    mv /tmp/conf_inicial.log /tmp/conf_inicial.log.1
    apt-get install -y apache2 php libapache2-mod-php libapache2-mod-php
php-mysql php-cli mariadb-server mariadb-client php-curl php-gd
php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip
libapache2-mpm-itk apt-utils jq certbot python3-certbot-apache >
```

```
/tmp/conf_inicial.log
    if [[ $? = 0 ]]; then
        echo -e "\t¡Hecho!\nSe han instalado los paqutes necesarios."
    else
        echo -e "¡Cachis! Parece que se ha producido un error. Revise
los logs en /tmp/conf_inicial.log"
        read -p "Pulsa cualquier tecla para continuar..."
        exit
    fi
    a2enmod rewrite >> /tmp/conf_inicial.log
    systemctl restart apache2 >> /tmp/conf_inicial.log
    cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
    wget -qO /etc/ssh/sshd_config
https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/
templates%20and%20misc/sshd_config >> /tmp/conf_inicial.log
    service ssh reload >> /tmp/conf_inicial.log
    mkdir -p /root/app_list/
    read -p "Pulsa cualquier tecla para continuar..."
    menu
}
```

config_wp.sh

```
function config_wp(){
    # Definición de variables
        db_name=wp_${1}
        db_user=${1}
        db_pw=${2}
    # Archivo wp-config.php
        # Copiar el archivo de ejemplo
        cp /var/www/${1}/blog/wp-config-sample.php
/var/www/${1}/blog/wp-config.php
        chown ${1}:${1} /var/www/${1}/blog/wp-config.php
        # Sustitución de valores
            sed -i "s/database_name_here/${db_name}/g"
"/var/www/${1}/blog/wp-config.php"
            sed -i "s/username_here/${1}/g"
"/var/www/${1}/blog/wp-config.php"
            sed -i "s/password_here/${2}/g"
"/var/www/${1}/blog/wp-config.php"

        #32:
https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/issues/32
        SALT=$(curl -L https://api.wordpress.org/secret-key/1.1/salt/)
        STRING='put your unique phrase here'
        printf '%s\n' "g/${STRING}/d" a "${SALT}" . w | ed -s
/var/www/${1}/blog/wp-config.php
```

```
}
```

crear_apache.sh

```
function crear_apache() {
    # Notas del fichero
        # VARS: Recibe ${usuario_nuevo} (${1})

    # Logging
        touch /var/log/apache2/${1}.${global_base_domain}.log
        touch /var/log/apache2/${1}.${global_base_domain}-access.log

        chmod 644 /var/log/apache2/${1}.${global_base_domain}.log
        chmod 644 /var/log/apache2/${1}.${global_base_domain}-access.log

        ln /var/log/apache2/${1}.${global_base_domain}.log
/var/www/${1}/ficheros/logs/${1}.${global_base_domain}.log

        ln /var/log/apache2/${1}.${global_base_domain}-access.log
/var/www/${1}/ficheros/logs/${1}.${global_base_domain}-access.log


    # Creación del sitio de Apache
        wget -qO /etc/apache2/sites-available/${1}.conf
https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/
templates%20and%20misc/virtualhost.txt
        sed -i "s/USER-TO-CHANGE/${1}/g"
"/etc/apache2/sites-available/${1}.conf"
        sed -i "s/GLOBAL-BASE-DOMAIN/${global_base_domain}/g"
"/etc/apache2/sites-available/${1}.conf"


    # Añadir página html para el sitio
        touch /var/www/${1}/web/index.html
        printf "Bienvenido al sitio del usuario ${1}" >
/var/www/${1}/web/index.html
        chown ${1}:${1} /var/www/${1}/web/index.html

    #Activar el sitio
        a2ensite ${1}.conf >> /dev/null
        systemctl reload apache2

    # Configuración ChrootDirectory y SSH
        cp /etc/ssh/sshd_config /etc/ssh/sshd_config_bak
        touch /tmp/sshd_config
        sed -r "s/^(Match User marcador.*$)/\1,${1}/"
"/etc/ssh/sshd_config" > /tmp/sshd_config
```

```
        mv /tmp/sshd_config /etc/ssh/sshd_config
}
```

crear_usuario.sh

```
function crear_usuario(){
    # Pido nombre de usuario. Paso a minúsculas. Reconfirmo
        read -p "Introduce el usuario a crear: " usuario_nuevo
        usuario_nuevo=${usuario_nuevo,,}
        echo -e "\e[5mAtención \e[0m: Se va a proceder a crear el
usuario \e[1m${usuario_nuevo}\e[0m"
        read -p "¿Es correcta la información? [s/n]: "
crear_apache_correct_user

    # Si no es correcto, salgo. Si existe, salgo
        if [[ ${crear_apache_correct_user} = "n" ]]; then
            echo -e "¡Recibido! \n Volviendo al menú. "
            menu
        fi

        egrep "^${usuario_nuevo}" /etc/passwd >/dev/null
        if [ ${?} -eq 0 ]; then
            echo "${usuario_nuevo} exists!"
            echo ""
            read -p "pulse cualquier tecla para continuar" caca
            menu
        fi
    # Crear carpetas
        echo -e "Creando usuario:  \e[1m${usuario_nuevo}\e[0m"
        mkdir -p /var/www/${usuario_nuevo}/{blog,web,ficheros}
        mkdir -p /var/www/${usuario_nuevo}/ficheros/logs
    # Genero las contraseñas y añado el usuario. Muestro la contraseña
        password_generada=$(openssl rand -base64 12)

        useradd -M -U --home /var/www/${usuario_nuevo} --shell /bin/bash
${usuario_nuevo}
        printf "${usuario_nuevo}:${password_generada}" | chpasswd

        echo "Usuario " ${usuario_nuevo} " creado"
        # echo "===============ANOTE================"
        # echo "||         LA CONTRASEÑA          ||"
        # echo "||                                ||"
        # echo "||                                ||"
        # echo "||      ${password_generada}          ||"
        # echo "||                                ||"
        # echo "||                                ||"
        # echo "===================================="
```

```
    # Modificar permisos y ownership
        chmod 755 /var/www/${usuario_nuevo}/
        chown -R ${usuario_nuevo}:${usuario_nuevo}
/var/www/${usuario_nuevo}/
        chown root:root /var/www/${usuario_nuevo}/
        chmod -R 770 /var/www/${usuario_nuevo}/*

    # Pausa de confirmación
        read -p "Pulse cualquier tecla para continuar " caca

    # Llamar a funciones extrañas
        crear_apache ${usuario_nuevo}
        #crear_wp ${usuario_nuevo} ${password_generada}
        #config_wp ${usuario_nuevo} ${password_generada}
        cf_updater ${usuario_nuevo}

        # Hasta que no se introduzca un email correcto, no se continúa
con la ejecución.
        read -p "Indique el correo electrónico del cliente: "
correo_cliente
        mail_regex="^[a-zA-Z0-9_-]+@[a-zA-Z_]+?\.[a-zA-Z]{2,12}$"
        until [[ ${correo_cliente} =~ ${mail_regex} ]];
        do
            echo -e "\e[5mERROR\e[0m: correo no válido.\n"
            read -p "Indique el correo electrónico del cliente: "
correo_cliente
        done

        envio_email ${usuario_nuevo} ${password_generada}
${correo_cliente} 1

        cert_creation ${usuario_nuevo}

    # Guardar contraseña por si fuera necesario.
        # No se utiliza por el momento
        #    save_passwd ${usuario_nuevo} ${password_generada}

    # Guardar el inventario de los recursos
        destination="/root/app_list/${usuario_nuevo}"

       echo "001" > ${destination}

    # Confirmación y menú
        echo -e "\nEl usuario ${usuario_nuevo} y sus sitios web se ha
creado correctamente. "
        read -rsp "Pulse cualquier tecla para continuar  " -n 1
        systemctl reload ssh
        menu
}
```

crear_wp.sh

```
function crear_wp(){
    # Notas del fichero
        # VARS: Recibe ${usuario_nuevo} (${1}) y ${password_generada}
(${2})
        # Lee de ".basrc" ${global_base_domain}
        # TODO: Poder crear solo un WP
        # TODO: Acceso remoto a la base de datos
        # TODO: PHPMyAdmin
        # CHECK: chown de los datos de WP. ¿www-data o user?

        mysql -e "CREATE DATABASE wp_${1};"
        mysql -e "CREATE USER '${1}'@localhost IDENTIFIED BY '${2}';"
        mysql -e "GRANT ALL PRIVILEGES ON wp_${1}.* TO
'${1}'@'localhost';"

        touch /var/log/apache2/blog.${1}.${global_base_domain}.log
        touch
/var/log/apache2/blog.${1}.${global_base_domain}-access.log

        chmod 644 /var/log/apache2/blog.${1}.${global_base_domain}.log
        chmod 644
/var/log/apache2/blog.${1}.${global_base_domain}-access.log

        ln /var/log/apache2/blog.${1}.${global_base_domain}.log
/var/www/${1}/ficheros/logs/blog.${1}.${global_base_domain}.log
        ln /var/log/apache2/blog.${1}.${global_base_domain}-access.log
/var/www/${1}/ficheros/logs/blog.${1}.${global_base_domain}-access.log

        wget -qO /etc/apache2/sites-available/wp_${1}.conf
https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/
templates%20and%20misc/wp_virtualhost
        sed -i "s/USER-TO-CHANGE/${1}/g"
"/etc/apache2/sites-available/wp_${1}.conf"
        sed -i "s/GLOBAL-BASE-DOMAIN/${global_base_domain}/g"
"/etc/apache2/sites-available/wp_${1}.conf"

        a2ensite wp_${1}.conf >> /dev/null
        sudo systemctl restart apache2

        if [[ ! -f /tmp/latest.tar.gz ]]; then
            curl https://wordpress.org/latest.tar.gz --output
/tmp/latest.tar.gz
            if [[  ! -d /tmp/wordpress ]]; then
                mkdir /tmp/wordpress &> /dev/null
```

```
                tar xzf /tmp/latest.tar.gz -C /tmp/wordpress
            fi
        fi

        cp -r /tmp/wordpress/wordpress/* /var/www/${1}/blog/
        chmod -R 770 /var/www/${1}/blog
        chown -R ${1}:${1} /var/www/${1}/blog
}
```

envio_email.sh

```
function envio_email() {
    # Notas del fichero
        # VARS: ${usuario_nuevo} (${1})
        # VARS: ${password_generada} (${2})
        # VARS: ${correo_cliente} (${3})
        # VARS: plantilla (${4})
          # Plantilla #1 --> Inicial
          # Plantilla #2 --> Cambio contraseña
          # Plantilla #3 --> Nuevo servicio (WP)
          # Plantilla #4 --> Nuevo servicio (PS)
        # VARS: Servicio nuevo (${5}):
          # blog --> WP
          # tienda --> PS
     plantilla=${4}

     if [[ ${plantilla} = 1 ]]; then
curl --request POST \
     --url https://api.sendinblue.com/v3/smtp/email \
     --header 'Accept: application/json' \
     --header 'Content-Type: application/json' \
     --header "api-key: "${global_sib_api_key}"" \
     --data '
{
     "to": [
          {
               "email": "'"${3}"'"
          }
     ],
     "params": {
          "SUBS_USERNAME": "'"${1}"'",
          "SUBS_PASSWORD": "'"${2}"'",
          "SUBS_HOST": "'"${1}"'",
          "SUBS_BASE_DOMAIN": "'"${global_base_domain}"'"
     },
     "templateId": 1
}
```

```
'
     elif [[ ${plantilla} = 2 ]]; then
curl --request POST \
     --url https://api.sendinblue.com/v3/smtp/email \
     --header 'Accept: application/json' \
     --header 'Content-Type: application/json' \
     --header "api-key: "${global_sib_api_key}"" \
     --data '
{
     "to": [
          {
               "email": "'"${3}"'"
          }
     ],
     "params": {
          "SUBS_USERNAME": "'"${1}"'",
          "SUBS_PASSWORD": "'"${2}"'"
     },
     "templateId": 2
}
'
     elif [[ ${plantilla} = 3 ]]; then
curl --request POST \
     --url https://api.sendinblue.com/v3/smtp/email \
     --header 'Accept: application/json' \
     --header 'Content-Type: application/json' \
     --header "api-key: "${global_sib_api_key}"" \
     --data '
{
     "to": [
          {
               "email": "'"${3}"'"
          }
     ],
     "params": {
          "SUBS_HOST": "'"${1}"'",
          "SUBS_SERVICE": "'"${5}"'",
          "SUBS_BASE_DOMAIN": "'"${global_base_domain}"'"
     },
     "templateId": 3
}
'
     elif [[ ${plantilla} = 4 ]]; then
curl --request POST \
     --url https://api.sendinblue.com/v3/smtp/email \
     --header 'Accept: application/json' \
     --header 'Content-Type: application/json' \
     --header "api-key: "${global_sib_api_key}"" \
     --data '
{
```

```
     "to": [
          {
               "email": "'"${3}"'"
          }
     ],
     "params": {
          "SUBS_USERNAME": "'"${1}"'",
          "SUBS_PASSWORD": "'"${2}"'",
          "SUBS_HOST": "'"${1}"'",
          "SUBS_SERVICE": "'"${5}"'",
          "SUBS_BASE_DOMAIN": "'"${global_base_domain}"'"
     },
     "templateId": 4
}
'
fi
}
```

install_prestashop.sh

```
    # VARS:
        # Usuario en el que configurar app
function install_prestashop(){
    echo -e "Instalando... \nEspere, por favor..."
    usuario=${1}
    mkdir -p /var/www/${usuario}/tienda
    chown ${usuario}:${usuario} /var/www/${usuario}/tienda
    chmod -R 770 /var/www/${usuario}/tienda

    touch /var/log/apache2/${usuario}.${global_base_domain}-tienda.log
    touch
/var/log/apache2/${usuario}.${global_base_domain}-tienda-access.log

    chmod 644
/var/log/apache2/${usuario}.${global_base_domain}-tienda.log
    chmod 644
/var/log/apache2/${usuario}.${global_base_domain}-tienda-access.log

    ln /var/log/apache2/${usuario}.${global_base_domain}-tienda.log
/var/www/${usuario}/ficheros/logs/${usuario}.${global_base_domain}-tiend
a.log
    ln
/var/log/apache2/${usuario}.${global_base_domain}-tienda-access.log
/var/www/${usuario}/ficheros/logs/${usuario}.${global_base_domain}-tiend
a-access.log
```

```
    wget -qO /etc/apache2/sites-available/tienda_${usuario}.conf
https://raw.githubusercontent.com/gonzaleztroyano/ASIR2-IAW-SCRIPT/main/
templates%20and%20misc/tienda_virtualhost.txt
    sed -i "s/USER-TO-CHANGE/${usuario}/g"
"/etc/apache2/sites-available/tienda_${usuario}.conf"
    sed -i "s/GLOBAL-BASE-DOMAIN/${global_base_domain}/g"
"/etc/apache2/sites-available/tienda_${usuario}.conf"
    a2ensite tienda_${usuario}.conf >> /dev/null
    systemctl restart apache2

    password_generada=$(openssl rand -base64 12)
    mysql -e "CREATE DATABASE tienda_${usuario};"
    mysql -e "CREATE USER '${usuario}_tienda'@localhost IDENTIFIED BY
'${password_generada}';"
    mysql -e "GRANT ALL PRIVILEGES ON tienda_${usuario}.* TO
'${usuario}_tienda'@'localhost';"

    if [[ ! -f /tmp/prestashop.zip ]]; then
        curl
https://download.prestashop.com/download/releases/prestashop_1.7.8.5.zip
--output /tmp/prestashop.zip
        if [[  ! -d /tmp/prestashop ]]; then
            mkdir /tmp/prestashop &> /dev/null
            unzip /tmp/prestashop.zip -d  /tmp/prestashop
        fi
    fi

    cp -r /tmp/prestashop/* /var/www/${usuario}/tienda/
    chmod -R 770 /var/www/${usuario}/tienda/
    chown -R ${usuario}:${usuario} /var/www/${usuario}/tienda/

    cf_updater ${usuario} tienda
    cert_creation "tienda.${usuario}"

    password_generada_user=$(openssl rand -base64 12)
    clear
    echo -e "\n========== ATENCIÓN ==========\n\n Acceda a:
https://tienda.${usuario}.${global_base_domain}\n\n========== GRACIAS
===========\n"
    read -p "¿Hecho? " trash

    clear
    echo -e "\n========== ATENCIÓN ==========\n\n Idioma: Español
(Spanish)\n\n========== GRACIAS ===========\n"
    read -p "¿Hecho? " trash

    clear
    echo -e "\n========== ATENCIÓN ==========\n\n Acepta términos y
pulsa 'Siguiente' \n\n========== GRACIAS ===========\n"
    read -p "¿Hecho? " trash
```

```
    clear
    echo -e "\n========== ATENCIÓN ==========\n\n Nombre: \nTienda de
${usuario}\n\n Actividad: Otra\n\n Datos demostración: 'Sí'\n\n Activar
SSL: Sí\n\n Correo del usuario: \n${usuario}@glez.tk\n\n Contraseña de
usuario:\n${password_generada_user}\n\n========== GRACIAS
===========\n"
    read -p "¿Hecho? " trash

    clear
    echo -e "\n========== ATENCIÓN ==========\n\n
BDD:\ntienda_${usuario}\n\n Usuario:\n${usuario}_tienda\n\n
Contraseña:\n${password_generada}\n\n========== GRACIAS ===========\n"
    read -p "¿Hecho? " trash
    clear
    echo "La tienda debería estar instalándose..."
    read -p "Pulsar al término de la instalación " trash

    rm -rf /var/www/${usuario}/tienda/install/

    mail_regex="^[a-zA-Z0-9_-]+@[a-zA-Z_]+?\.[a-zA-Z]{2,12}$"
    until [[ ${correo_cliente} =~ ${mail_regex} ]];
    do
        echo -e "\e[5mERROR\e[0m: correo no válido.\n"
        read -p "Indique el correo electrónico del cliente: "
correo_cliente
    done
    envio_email ${usuario} ${password_generada_user} ${correo_cliente} 4
tienda

}
```

listar.sh

```
function listar() {

    salir=0

    while [ ${salir} != 1 ]
    do
        echo ""
        read -p "¿Desea buscar algún nombre de usuario en concreto?
[s/N]: " buscar_usuario_filtro

        if [[ ${buscar_usuario_filtro} = "s" ]]; then

            read -p "Introduzca el término a buscar: "
```

```
buscar_usuario_filtro_termino
            echo -e "Estos son los usuarios que coinciden con el término
indicado: \n "
            cat /etc/passwd | grep '/var/www' | grep
${buscar_usuario_filtro_termino} | cut -d ':' -f 1
            echo -e "\n -- FIN DE LA LISTA -- \n "
            salir=1

        else
            echo -e "Usuarios del sistema web: \n "
            cat /etc/passwd | grep '/var/www' | cut -d ':' -f 1
            echo -e "\n -- FIN DE LA LISTA -- \n "
            salir=1
        fi
    done
    read -p "Pulse cualquier tecla para volver al menú." caca
    menu
}
```

menu.sh

```
function menu() {

    show_header

        echo ""
        echo ""
        echo "  1. Listar usuarias"
        echo "  2. Crear usuarios"
        echo "  31. Añadir aplicación a un usuario"
        echo "  32. Ver las aplicaciones de un usuario"
        echo "  4. Borrar usuarias"
        echo "  5. Modificar usuarios"
        echo "  6. Salir del programa"
        echo ""
        echo "  7. Configuración de secretos"
        echo "  8. Configuración inicial del servidor."
        echo "  9. Borrar usuario directamente (sin preguntar)"
        echo ""
        read -p "     Opción seleccionada: " seleccionada

        if [[ ${seleccionada} = 1 ]]; then
                listar
            elif [[ ${seleccionada} = 2 ]]; then
                crear_usuario
            elif [[ ${seleccionada} = 31 ]]; then
                add_app
```

```
            elif [[ ${seleccionada} = 32 ]]; then
                app_list bonito
            elif [[ ${seleccionada} = 4 ]]; then
                borrar
            elif [[ ${seleccionada} = 5 ]]; then
                modificar
            elif [[ ${seleccionada} = 6 ]]; then
                echo ""
                exit
            elif [[ ${seleccionada} = 7 ]]; then
                secrets
            elif [[ ${seleccionada} = 8 ]]; then
                conf_inicial
            elif [[ ${seleccionada} = 9 ]]; then
                borrar_hard
            else
                echo "Opción no válida"
                menu
        fi
}
```

modificar.sh

```
function modificar(){
    # Listar usuarios
        echo -e "Usuarios del sistema web: \n "
            cat /etc/passwd | grep '/var/www' | cut -d ':' -f 1
            echo -e "\n -- FIN DE LA LISTA -- \n \n"

    # Pedir usuario a modificar
        read -p "¿Qué usuario deseas modificar? " usuario_a_modificar

    # Comprobar si el usuario existe
        check_usuario_existe=$(cat /etc/passwd | grep "/var/www" | cut
-d ":" -f 1 | grep -w ${usuario_a_modificar})
        if [[ ${check_usuario_existe} != ${usuario_a_modificar} ]]; then

            read "El usuario indicado no existe" caca
            menu
        else

    # Si existe, actuar:
        # Pedir contraseña nueva, dos veces por seguridad
            read -p "Introduce una nueva contraseña para el usuario
${usuario_a_modificar}: " password_nueva_1
            read -p "Introduce de nuevo la contraseña para el usuario
${usuario_a_modificar}: " password_nueva_2
```

```
        # Comparar contraseñas
            if [[ ${password_nueva_1} = ${password_nueva_2} ]]; then
                    printf "${usuario_a_modificar}:${password_nueva_1}"
| chpasswd
                    echo "¡Contraseña actualizada!"
                    read -p "Pulse cualquier tecla para continuar" caca
            else
                echo "\e[5mERROR \e[0m: las contraseñas no coinciden"
                read -p "Pulse cualquier tecla para continuar" caca
                menu
            fi
        fi
        #TODO: #39 Enviar correo con la contraseña cambiada.
            # Hasta que no se introduzca un email correcto, no se
continúa con la ejecución.
            read -p "Indique el correo electrónico del cliente: "
correo_cliente
            mail_regex="^[a-zA-Z0-9_-]+@[a-zA-Z_]+?\.[a-zA-Z]{2,12}$"
            until [[ ${correo_cliente} =~ ${mail_regex} ]];
            do
                echo -e "\e[5mERROR\e[0m: correo no válido.\n"
                read -p "Indique el correo electrónico del cliente: "
correo_cliente
            done
            envio_email ${usuario_a_modificar} ${password_nueva_1}
${correo_cliente} 2
            echo "¡Listo!"
            read -p "Pulse cualquier tecla para volver al menú" caca
    menu
}
```

save_passwd.sh (Sin uso)

```
function save_passwd(){
     # Notas del fichero
        # VARS: username (${1})
        # VARS: password (${2})

    destination="/root/user_credentials/${username}"

    if [ ! -f ${destination} ]; then
        touch ${destination}
    fi

    echo ${1} >> ${destination}
}
```

secrets.sh

```
    # Notas del fichero
        # Related issue: #21
function secrets() {
    show_header
    echo -e "\e[1mIntroduzca los secretos y variables solicitadas\e[0m"
    echo ""
    read -rp "[1/5] - Dominio base para la configuración: " base_domain
    echo -e "\tGuardado. \n"
    read -rp "[2/5] - Email de la cuenta en Cloudflare: " cf_email
    echo -e "\tGuardado. \n"
    read -rp "[3/5] - ID de zona en Cloudflare: " cf_zone
    echo -e "\tGuardado. \n"
    read -rp "[4/5] - Token API de Cloudflare: " cf_token
    echo -e "\tGuardado. \n"
    read -rp "[5/5] - Token API de SendInBlue: " sib_api_key

    {
    echo "global_base_domain=\"${base_domain}\""
    echo "global_cf_email=\"${cf_email}\""
    echo "global_cf_zone=\"${cf_zone}\""
    echo "global_cf_token=\"${cf_token}\""
    echo "global_sib_api_key=\"${sib_api_key}\""
    echo "export global_base_domain"
    echo "export global_cf_email"
    echo "export global_cf_zone"
    echo "export global_cf_token"
    echo "export global_sib_api_key"
    } >> ~/.bashrc

    export global_base_domain=\"${base_domain}\"
    export global_cf_email=\"${cf_email}\"
    export global_cf_zone=\"${cf_zone}\"
    export global_cf_token=\"${cf_token}\"
    export global_sib_api_key=\"${sib_api_key}\"

    echo -e "\n \nSe han guardado los secretos.\nEs posible que debas
reiniciar la sesión para ver aplicados los cambios."
    read -p "Pulse cualquer tecla para continuar" trash
    menu
}
```

show_header.sh

```
function show_header() {
        clear
        echo ""
        echo ""
cat << "EOF"


   _____  _____ _____  _____ _____ _______   _____  ______    _____ ______  _____ _______ _____ ____  _   _ 
  / ____|/ ____|  __ \|_   _|  __ \__   __| |  __ \|  ____|  / ____|  ____|/ ____|__   __|_   _/ __ \| \ | |
 | (___ | |    | |__) | | | | |__) | | |    | |  | | |__    | |  __| |__  | (___    | |    | || |  | |  \| |
  \___ \| |    |  _  /  | | |  ___/  | |    | |  | |  __|   | | |_ |  __|  \___ \   | |    | || |  | | . ` |
  ____) | |____| | \ \ _| |_| |      | |    | |__| | |____  | |__| | |____ ____) |  | |   _| || |__| | |\  |
 |_____/ \_____|_|  \_\_____|_|      |_|    |_____/|______|  \_____|______|_____/   |_|  |_____\____/|_| \_|
                              CC BY 4.0 Internacional Pablo González
                       https://github.com/gonzaleztroyano/ASIR2-IAW-SCRIPT



EOF

}
```

sshd_config

```
# En GCP no cambiamos. Sí en otros proveedores. Confiugrar SSH Keys
#Port 2225
PermitRootLogin no
LoginGraceTime 60
Subsystem sftp internal-sftp
PrintMotd no
SyslogFacility AUTH
LogLevel INFO
MaxAuthTries 2
MaxSessions 2
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM yes
#X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*
ClientAliveInterval 120
UseDNS no

Match User marcador
    ChrootDirectory %h
```

```
    ForceCommand internal-sftp -u 0027
    PasswordAuthentication yes
```

tienda_virtualhost.txt

```
<VirtualHost *:80>
    ServerAdmin admin@localhost
    ServerName tienda.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN
    DocumentRoot /var/www/USER-TO-CHANGE/tienda

    <Directory /var/www/USER-TO-CHANGE/tienda>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog
/var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-tienda.log
    CustomLog
/var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-tienda-access.log
combined
    AssignUserID USER-TO-CHANGE USER-TO-CHANGE

</VirtualHost>
```

virtualhost.txt

```
<VirtualHost *:80>
    ServerName USER-TO-CHANGE.GLOBAL-BASE-DOMAIN
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/USER-TO-CHANGE/web
    ErrorLog /var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN.log
    CustomLog
/var/log/apache2/USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-access.log combined
    AssignUserID USER-TO-CHANGE USER-TO-CHANGE
</VirtualHost>
```

wp_virtualhost

```
<VirtualHost *:80>
    ServerAdmin USER-TO-CHANGE@localhost
    ServerName blog.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN
```

```
    DocumentRoot /var/www/USER-TO-CHANGE/blog
    ErrorLog /var/log/apache2/blog.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN.log
    CustomLog
/var/log/apache2/blog.USER-TO-CHANGE.GLOBAL-BASE-DOMAIN-access.log
combined
    AssignUserID USER-TO-CHANGE USER-TO-CHANGE
    <Directory /var/www/USER-TO-CHANGE/blog/>
        AllowOverride All
    </Directory>
</VirtualHost>
```

cloudflare-cleaner.py

```
#!/bin/env python3

import CloudFlare
import os
import sys

def main():
    try:
        zone_name = sys.argv[1]
        dns_name = sys.argv[2]
    except IndexError:
        exit('usage: example_delete_zone_entry.py zone dns_record')

    cf = CloudFlare.CloudFlare(token='TOKEN')
    try:
        params = {'name':zone_name}
        zones = cf.zones.get(params=params)
    except CloudFlare.exceptions.CloudFlareAPIError as e:
        exit('/zones %d %s - api call failed' % (e, e))
    except Exception as e:
        exit('/zones.get - %s - api call failed' % (e))

    if len(zones) == 0:
        exit('/zones.get - %s - zone not found' % (zone_name))

    if len(zones) != 1:
        exit('/zones.get - %s - api call returned %d items' %
(zone_name, len(zones)))

    zone = zones[0]
    zone_id = zone['id']
    zone_name = zone['name']

    print('ZONE:', zone_id, zone_name)
    try:
```

```
        params = {'name':dns_name + '.' + zone_name}
        dns_records = cf.zones.dns_records.get(zone_id, params=params)
    except CloudFlare.exceptions.CloudFlareAPIError as e:
        exit('/zones/dns_records %s - %d %s - api call failed' %
(dns_name, e, e))

    found = False
    for dns_record in dns_records:
        dns_record_id = dns_record['id']
        dns_record_name = dns_record['name']
        dns_record_type = dns_record['type']
        dns_record_value = dns_record['content']
        print('DNS RECORD:', dns_record_id, dns_record_name,
dns_record_type, dns_record_value)
        try:
            dns_record = cf.zones.dns_records.delete(zone_id,
dns_record_id)
            print('DELETED')
        except CloudFlare.exceptions.CloudFlareAPIError as e:
            exit('/zones.dns_records.delete %s - %d %s - api call
failed' % (dns_name, e, e))
        found = True

    if not found:
        print('RECORD NOT FOUND')

    exit(0)

if __name__ == '__main__':
    main()
```

7.8. Anexo VIII: Códigos relativos a soluciones Out-of-the-box

correo-educa2Plesk.txt

```
Return-Path: <pablo\.\gonzalez XXX [[@]]] educa\.\madrid\.\org>
X-Original-To: info[[@]]xn--ahorrams-fza.es
Delivered-To: info[[@]]xn--ahorrams-fza.es
Received: from mx01.puc.rediris.es (outbound5mad.lav.puc.rediris.es [130.206.19.148])
       by compassionate-curran.5-175-45-212.plesk.page (Postfix) with ESMTPS id
92E4982F93
       for <info[[@]]xn--ahorrams-fza.es>; Fri,  6 May 2022 13:03:30 +0200 (CEST)
Authentication-Results: plesk.glez.cloud;
       dmarc=pass (p=REJECT sp=NONE) smtp.from=educa.madrid.org
header.from=educa.madrid.org;
       dkim=pass header.d=educa.madrid.org;
       spf=pass (sender IP is 130.206.19.148) smtp.mailfrom=pablo\.\gonzalez XXX [[@]]]
educa\.\madrid\.\org smtp.helo=mx01.puc.rediris.es
Received-SPF: pass (plesk.glez.cloud: domain of educa.madrid.org designates
130.206.19.148 as permitted sender) client-ip=130.206.19.148;
envelope-from=pablo\.\gonzalez XXX [[@]]] educa\.\madrid\.\org;
helo=mx01.puc.rediris.es;
Authentication-Results: mx01.puc.rediris.es;
       spf=pass (rediris.es: domain of pablo\.\gonzalez XXX [[@]]] educa\.\madrid\.\org
designates 193.146.123.99 as permitted sender) smtp.mailfrom=pablo\.\gonzalez XXX [[@]]]
educa\.\madrid\.\org
Received: from smtp.educa.madrid.org ([193.146.123.99])
       by mx01.puc.rediris.es  with ESMTP id 246B3ULo011528-246B3ULp011528
       for <info[[@]]xn--ahorrams-fza.es>; Fri, 6 May 2022 13:03:30 +0200
Received: (qmail 7712 invoked from network); 6 May 2022 11:03:29 -0000
Received: from emvrcubeweb01.educa.madrid.org (HELO WEBMAIL) ([172.16.2.181])
          (envelope-sender <pablo\.\gonzalez XXX [[@]]] educa\.\madrid\.\org>)
          by 0 (qmail-ldap-1.03) with SMTP
          for <info@xn--ahorrams-fza.es>; 6 May 2022 11:03:29 -0000
MIME-Version: 1.0
Date: Fri, 06 May 2022 13:03:29 +0200
From: =?UTF-8?Q?Pablo_Gonz=C3=A1lez_Troyano?=
 <pablo\.\gonzalez XXX [[@]]] educa\.\madrid\.\org>
To: info[[@]]xn--ahorrams-fza.es
Subject: PRUEBA ENVIO
User-Agent: CorreoWeb EducaMadrid
Message-ID: <17697d3c5c33f722e6b458604c9fbf8f@educa.madrid.org>
X-Sender: pablo\.\gonzalez XXX [[@]]] educa\.\madrid\.\org
Organization: Comunidad de Madrid. EducaMadrid.
X-Remote-Browser: Mozilla/5.0 (X11; CrOS x86_64 14526.89.0) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/100.0.4896.133 Safari/537.36
X-Originating-IP: [79.116.1.157]
X-Webmail-Server: 172.16.2.181
Content-Type: multipart/alternative;
 boundary="=_c6f958ea275aa61f234210d1ed75fe93"
X-FE-Policy-ID: 23:14:2:educa.madrid.org
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; d=educa.madrid.org; s=dkim_educamadrid;
c=relaxed/relaxed;
 h=mime-version:date:from:to:subject:message-id:content-type;
 bh=p+I+BArBMd8dSkDKaCCFDj1/nkdqWHoUR+1qCg1YroU=;
 b=oe+fsPmgJa7Km5FbPgAvVHTB6BKEeB3GbdljSu7LRfCzam2aYcKE4kzqdDlM7/DwGTJc8a0Nx51f
       4TaK718n6do5ov9EmkKK68iiWVlgb8NV0V7lA2ri6t3slyozlTTfqo4/xFWEFBA3edYdDjugAHst
       8V8EHfh2FU6YAfw5fIyLJyIoUUd4TcDe3nAj04Wa+KmiRcy3Ws+o4myJ1nOZZt26eAjXhi0f3MsB
       Tp+LTK0Llk8Q+eIQ9rXDrLq2LU/+qzZDlresx95kosmILaabnKxuLlC4BJR/hzVDinzxnRn3pXo1
       TE3HimttUgG/Rpjd56e0aBqbhpAVa8LUHgGyyA==

--=_c6f958ea275aa61f234210d1ed75fe93
Content-Transfer-Encoding: 8bit
```

```
Content-Type: text/plain; charset=UTF-8;
 format=flowed

HOLAAA

---
PABLO GONZÁLEZ TROYANO
pablo\.\gonzalez XXX [[@]]] educa\.\madrid\.\org
2º ASIR
IES Villablanca, Madrid
--=_c6f958ea275aa61f234210d1ed75fe93
Content-Transfer-Encoding: quoted-printable
Content-Type: text/html; charset=UTF-8

<html><head><meta http-equiv=3D"Content-Type" content=3D"text/html; charset=
=3DUTF-8" /></head><body style=3D'font-size: 10pt; font-family: Arial,Helve=
tica,sans-serif'>
<p>HOLAAA</p>
<div id=3D"signature">
<div class=3D"pre" style=3D"margin: 0; padding: 0; font-family: monospace">=
---<br />PABLO GONZ&Aacute;LEZ TROYANO<br /><a href=3D"mailto:pablo\.\gonzale=
z [[@]]] educa\.\madrid\.\org">pablo\.\gonzalez [[@]]] educa\.\madrid\.\org</a><br
/>2&ordm; =
ASIR<br />IES Villablanca, Madrid</div>
</div>
</body></html>

--=_c6f958ea275aa61f234210d1ed75fe93--
```

correo-Plesk2Educa.txt

```
Received: (qmail 26254 invoked by uid 7007); 6 May 2022 11:09:07 -0000
Received: from unknown ([192.168.2.109])
          by 0 (qmail-ldap-1.03) with QMQP; 6 May 2022 11:09:07 -0000
Delivered-To: CLUSTERHOST emvav01.educa.madrid.org
pablo\.gonzalezXXX[[@]]educa\.\madrid\.org
Received: (qmail 20398 invoked from network); 6 May 2022 11:09:07 -0000
Received: from unknown (HELO mx01.puc.rediris.es) ([130.206.19.148])
          (envelope-sender <info[[@]]xn--ahorrams-fza.es>)
          by 0 (qmail-ldap-1.03) with SMTP
          for <pablo\.gonzalezXXX[[@]]educa\.\madrid\.org>; 6 May 2022 11:09:07 -0000
Authentication-Results: mx01.puc.rediris.es;
       spf=pass (rediris.es: domain of info[[@]]xn--ahorrams-fza.es designates
5.175.45.212 as permitted sender) smtp.mailfrom=info[[@]]xn--ahorrams-fza.es
       dmarc=pass header.from=xn--ahorrams-fza.es
Received: from compassionate-curran.5-175-45-212.plesk.page (glez-cloud.vservers.es
[5.175.45.212])
       by mx01.puc.rediris.es  with ESMTP id 246B95ng017592-246B95ni017592
       (version=TLSv1.3 cipher=TLS_AES_256_GCM_SHA384 bits=256 verify=NO)
       for <pablo\.gonzalezXXX[[@]]educa\.\madrid\.org>; Fri, 6 May 2022 13:09:05 +0200
Received: from webmail.xn--ahorrams-fza.es (localhost.localdomain [IPv6:::1])
       by compassionate-curran.5-175-45-212.plesk.page (Postfix) with ESMTPSA id
EA16782F93;
       Fri,  6 May 2022 13:09:04 +0200 (CEST)
Authentication-Results: plesk.glez.cloud;
        spf=pass (sender IP is ::1) smtp.mailfrom=info[[@]]xn--ahorrams-fza.es
smtp.helo=webmail.xn--ahorrams-fza.es
Received-SPF: pass (plesk.glez.cloud: connection is authenticated)
MIME-Version: 1.0
```

```
Date: Fri, 06 May 2022 13:09:04 +0200
From: info[[@]]xn--ahorrams-fza.es
To: =?UTF-8?Q?Pablo_Gonz=C3=A1lez_Troyano?=
 <pablo\.gonzalezXXX[[@]]educa\.\madrid\.org>
Cc: info[[@]]xn--ahorrams-fza.es
Subject: Re: PRUEBA ENVIO
In-Reply-To: <17697d3c5c33f722e6b458604c9fbf8f[[@]]educa.madrid.org>
References: <17697d3c5c33f722e6b458604c9fbf8f[[@]]educa.madrid.org>
User-Agent: Roundcube Webmail/1.4.13
Message-ID: <a73d28e23c448311b7a6f4817327a07f[[@]]xn--ahorrams-fza.es>
X-Sender: info[[@]]xn--ahorrams-fza.es
Content-Type: text/plain; charset=UTF-8;
 format=flowed
Content-Transfer-Encoding: 8bit
X-FEAS-SPF: spf-result=pass, ip=5.175.45.212,
helo=compassionate-curran.5-175-45-212.plesk.page, mailFrom=info[[@]]xn--ahorrams-fza.es
X-FE-Policy-ID: 21:3:1:educa.madrid.org

Buenas, buenas!

El 2022-05-06 13:03, Pablo González Troyano escribió:
> HOLAAA
>
> ---
> PABLO GONZÁLEZ TROYANO
> pablo\.gonzalezXXX[[@]]educa\.\madrid\.org
> 2º ASIR
> IES Villablanca, Madrid
```

7.9. Anexo IX: Seguridad en el correo electrónico: DKIM, SPF y DMARC

DKIM (*DomainKeys Identified Mail*), SPF (*Sender Policy Framework*) y DMARC (*Domain-based Message Authentication, Reporting and Conformance*) son protocolos desarrollados por la comunidad técnica que tiene como objetivo prevenir el spoofing de correo electrónico, asegurando la entrega de los mensajes adecuados y el marcado de como spam de los mensajes no deseados.

Si bien estos tres protocolos están diseñados para trabajar de forma conjunta, el más opcional de todos es DMARC. DKIM y SPF se pueden configurar de forma independiente.

Antes de aplicar cualquier cambio que pueda afectar a la distribución de los mensajes de correo electrónico es recomendable realizar un estudio de los efectos que se pueden producir. Una mala o incompleta configuración puede provocar que correos enviados por nuestros usuarios sean rechazados o marcados como Spam.

Para configurarlos necesitamos acceso a editar los registros DNS de nuestro dominio, pues todos se añaden como entradas TXT.

En tanto a **DKIM**, este protocolo se encarga de firmar digitalmente los correos enviados desde nuestras cuentas. Permite que si un mensaje ha sido modificado en tránsito (mientras viajan entre servidores) no pase desapercibido; además sirve para comprobar que el correo proviene del dominio del que dice proceder.

Sin entrar en detalles técnicos, utiliza una clave asimétrica (clave pública - clave privada) para firmar los mensajes. Son firmados utilizando la clave privada, que gestiona el proveedor, y el servidor de correo que recibe el mensaje comprueba (utilizando la clave pública que se encuentra publicada como TXT en nuestro dominio) si concuerdan.

En el caso del dominio gonzaleztroyano.es, que tiene el servicio de correo electrónico en Google Workspace podemos ver el siguiente registro:

```
dig TXT google._domainkey.gonzaleztroyano.es +short
```

```
"v=DKIM1; k=rsa;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAgFzQIWqfpq2XbSazxHsoIBaYnE9gJ04X9kI2jYdLYJ
WtIUZD9wsJ1jY2K5Esdd768lWhz5APXnDRuAeTsupAx/h9QleMzWSna3khC01YQuw4FyxllKUoqbSb3C7atm79Qb
J9VXiqzCwc/0LzI73d1sPkBwRVcy5Nii4GHHZGzQ3vRHvBRY2angrqPsN3s1fX0"
"nQCeaH8L4r/aTALKN1xx5WZ61BfLcwd4gc70vpK/ABlaxXePlI0FifoFCeoxH9A6SRlZF75EcE8RWZmfI/0LIEo
XNXsOwsyBmIfCx6/OU5gN+zdqx3vIwSF5FpDaJHdLsWzS9fRZp9qdwos4OKPcwIDAQAB"
```

En el caso de los correos del subdominio educa.madrid.org se usa el siguiente selector DKIM para la firma de los correos electrónicos salientes:

```
dig TXT dkim_educamadrid._domainkey.educa.madrid.org +short
```

```
"v=DKIM1; k=rsa; p="
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAr9MeSYdTKbW2Sa+Iwizf1dJEiqYxNKG1WtgqfU92jF5
SU1urPxZy3FBfq5qZtGbsXL6TcTNAXUOwKv9D3r5C6ddZSrfVaNMRfHiMGJ8xZuLHSSXerN89zh3wPKA4T6A+3zW
2igoy3JAk7i8ZfjOuqiqO+iDe" "a6ViwTgevG+g0ut3f3GtkpwdK4GNOWt"
"rPoplDuMalXl0kGRjpIagGy2d2v/sX5Yr5om+veJb2XXHbBaMk7g8w5V7nCE6XtnXWneOyLoaptyZK4/mbzA+IN
FdL1ZEEedla7Gx+d+QPHobr2SMikAvN1hixMYdn+yW5/pz5d3e4R4azlP/JK7zJsFfOwIDAQAB"


# NOTA: Se ven varios registros, encerrados entre comillas (") debido a los límites que
algunos servidores DNS imponen en la longitud máxima de los registros TXT
```

Es posible que un dominio tenga varios selectores DKIM, cada uno de un servicio o servidor. Para ver con qué selector se cifra cada correo, podemos investigar las cabeceras a la hora de recibirlo. En el caso del ejemplo anterior:

Sobre **SPF**, es utilizado para identificar desde qué servidores (por sus direcciones IP) esperamos como propietarios del dominio que se envíen mensajes en nuestro nombre.

En el caso del dominio gonzaleztroyano.es, vemos el siguiente registro SPF. Tal y como se muestra en el registro, solo Google y Sendinblue están autorizados para enviar correo electrónico utilizando este dominio.

```
dig TXT gonzaleztroyano.es +short | grep spf1
```

```
"v=spf1 include:_spf.google.com include:spf.sendinblue.com -all"
```

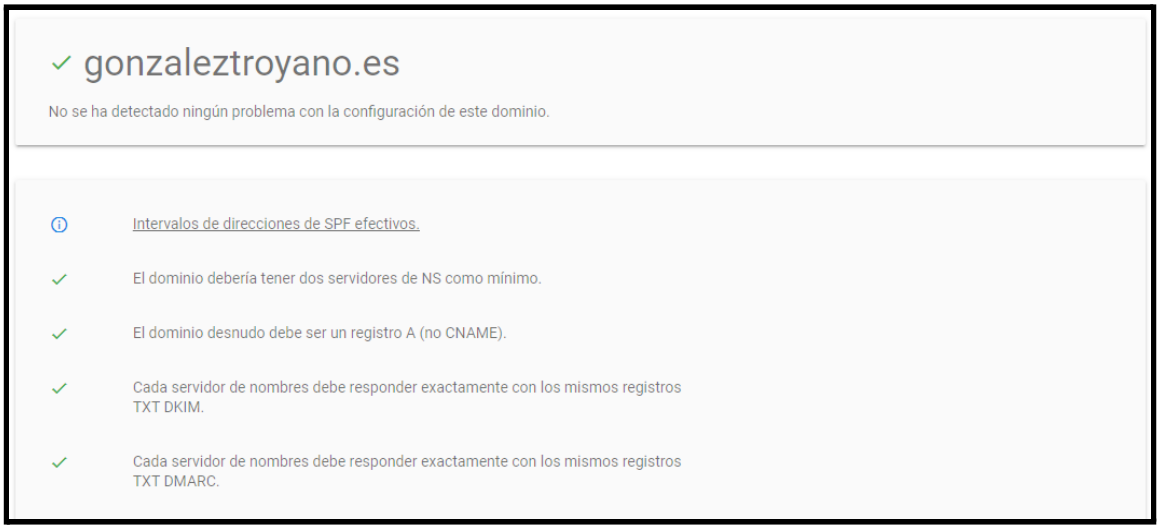
En tanto al registro SPF de EducaMadrid, vemos que autoriza una serie de IPs en REDIRIS y otro par de ellas más. Al igual que en gonzaleztroyano.es, el "-all" deniega todos los orígenes que no estén explícitamente indicados en el registro

```
dig TXT educa.madrid.org +short | grep spf1
"v=spf1 include:spf.puc.rediris.es ip4:193.146.123.99 ip4:193.146.123.95 -all"
```

DMARC es el último de los protocolos que aquí trataremos. Es el protocolo que cierra el círculo en la seguridad del correo electrónico, y no es estrictamente necesaria su configuración (la implementación de SPF y DKIM sí es altamente recomendable).

Utilizándolo podemos anunciar la política que queremos que otros sistemas utilicen en el caso de que reciban un mensaje de correo electrónico que aparentemente proceda de nuestro dominio pero que falle en alguno de los protocolos anteriormente descritos.

Para comprobar la correcta aplicación de la política podemos usar esta herramienta online[123] que pone Google a disposición de administradores. No es necesario ningún tipo de registro y aunque la información no es muy exhaustiva, para un primer reconocimiento puede ser muy útil.



124

[123] https://toolbox.googleapps.com/apps/checkmx/
[124] https://toolbox.googleapps.com/apps/checkmx/check?domain=gonzaleztroyano.es&dkim_selector=

8. Monitorización y visibilidad sobre infraestructura

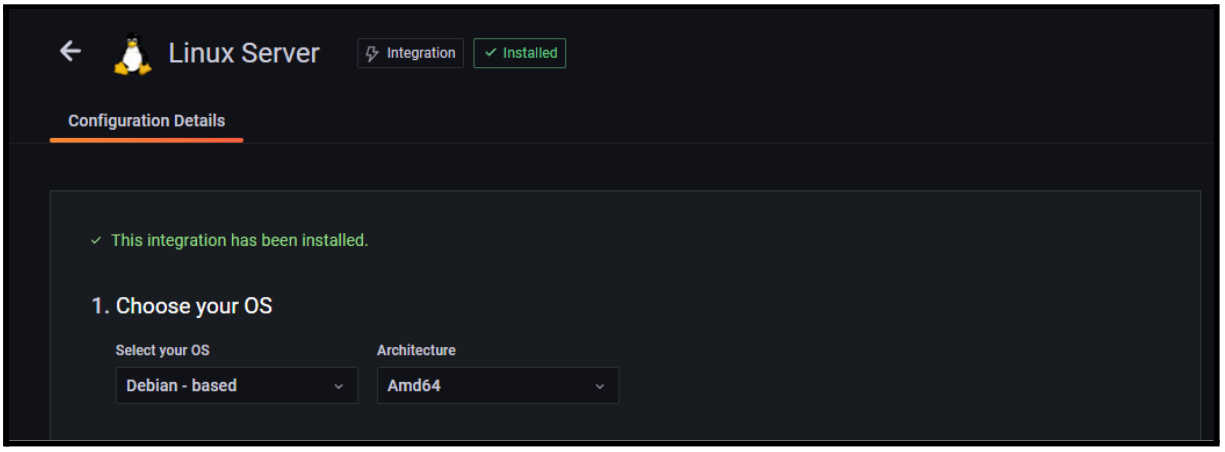
8.1. Instalación del agente

Tal y como se ha explicado anteriormente, se usa Grafana para la gestión de logs, métricas, monitorización y alertas de estado.

Se instala el agente de monitorización de Grafana[125] en todos los nodos. Este agente de monitorización recolecta información sobre CPU, memoria, red, etc. Además, envía los logs a Grafana de forma automática.

Además, envía los logs a Grafana de forma automática.

La instalación es sencilla. Desde el panel web de Grafana, nos situamos sobre *Integrations and Connections*. Entre todas las opciones, seleccionamos *Linux Server*. Escogemos nuestra arquitectura:



Ejecutamos el siguiente comando para ejecutar el script que instalará y configurará el servicio:

```
sudo ARCH=amd64 GCLOUD_STACK_ID="260642" GCLOUD_API_KEY="[ESTO ES SECRETO]"
GCLOUD_API_URL="https://integrations-api-eu-west.grafana.net" /bin/sh -c "$(curl
-fsSL
https://raw.githubusercontent.com/grafana/agent/release/production/grafanacloud-
install.sh)"
```

[125] https://grafana.com/oss/prometheus/exporters/node-exporter/

8.2. Monitorización de bases de datos

En el caso de los equipos que tienen instalada una base de datos (como son web-server-glez-cloud y plesk.glez.cloud), se configura la integración con los pasos descritos a continuación:

```
mysql # Para acceder a la CLI de MariaDB/mysql

CREATE USER 'READ_ONLY_udijPM97sgUUubkTXAg1PtiF'@'localhost' IDENTIFIED BY
'SbhE1VyNQSmG9qDMKxIu5YtNtdwxdvwn';

GRANT SELECT ON *.* TO 'READ_ONLY_udijPM97sgUUubkTXAg1PtiF'@'localhost'
IDENTIFIED BY 'SbhE1VyNQSmG9qDMKxIu5YtNtdwxdvwn';

FLUSH PRIVILEGES;
```

Contraseñas generadas con este generador[126] de contraseñas funcionando en Cloudflare Workers[127]. Más información en este artículo de blog[128].

Una vez creado el usuario que usará el agente de grafana para autenticarse contra el servidor de bases de datos, debemos editar el archivo de configuración del propio agente, ubicado en la ruta `/etc/grafana-agent.yaml`.

Aquí modificaremos la siguiente línea de información desde esto:

```
integrations:
  mysqld_exporter:
    data_source_name: root@(localhost:3306)/
```

A esta otra línea con la información de acceso ya actualizada:

```
integrations:
  mysqld_exporter:
    data_source_name:
READ_ONLY_udijPM97sgUUubkTXAg1PtiF:SbhE1VyNQSmG9qDMKxIu5YtNtdwxdvwn@(localhost:3
306)/
```

Se modifica también el nombre de la instancia, para reconocerla más fácilmente desde Grafana a

```
instance: mysql-plesk-glez-cloud
```

[126] https://genera-password.gonzaleztroyano.es/
[127] https://workers.cloudflare.com/
[128] https://blog.gonzaleztroyano.es/cloudflare-workers-password-generator/

8.3. Monitorización de servicios con Grafana

Se han definido los siguientes *endpoints* para monitorizar usando Grafana[129]:

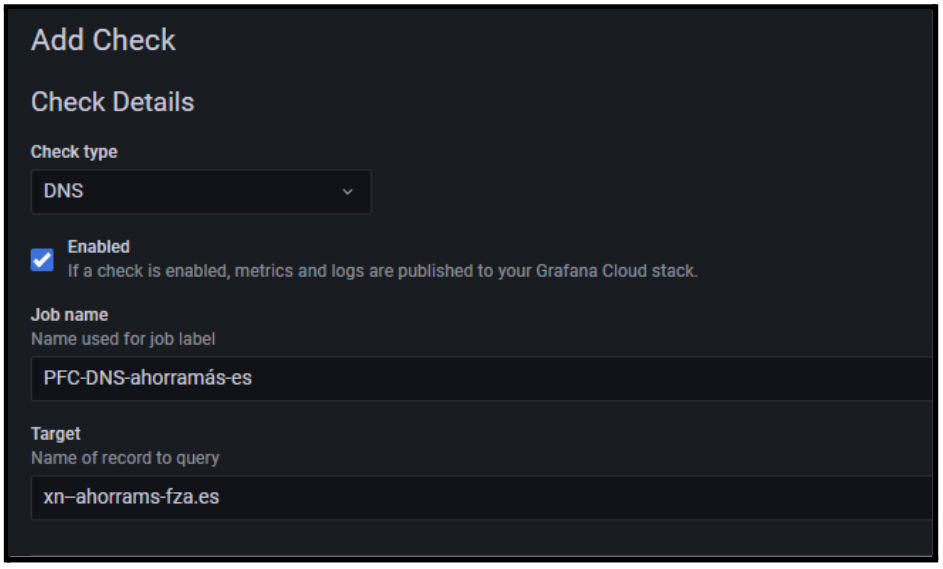
#	Check Name	Target	Type	Frequency	Labels
1	PFC-PING-web-server-glez-cloud	web-server.glez.cloud	PING	120	pfc:True server:web-server-oci provider:oci
2	PFC-PING-plesk-glez-cloud	plesk.glez.cloud	PING	120	pfc:True server:plesk-axarnet provider:axarnet
3	PFC-PING-ns1-glez-cloud	ns1.glez.cloud	PING	120	pfc:True server:ns1-clouding provider:clouding
4	PFC-PING-mail-glez-cloud	mail.glez.cloud	PING	120	pfc:True server:mail-clouding provider:clouding
5	PFC-PING-voip-glez-cloud	voip.glez.cloud	PING	120	pfc:True server:voip-clouding provider:clouding
6	PFC-DNS-Ahorramas-es	xn--ahorrams-fza.es	DNS	120	pfc:True server:plesk-axarnet provider:axarnet
7	PFC-DNS-Ahorramas-com	xn--ahorrams-fza.com	DNS	120	pfc:True server:ns1-clouding provider:clouding
8	PFC-HTTP-ayuda-glez-cloud	https://ayuda.glez.cloud	HTTP	120	pfc:True server:plesk-axarnet provider:axarnet
9	PFC-HTTP-glez-cloud	https://glez.cloud	HTTP	120	pfc:True server:plesk-axarnet provider:axarnet
10	PFC-HTTP-eat-a-lot-glez-cloud	https://eat-a-lot.glez.cloud/wp-content/uploads/2022/04/cropped-cropped-verde-pistacho-2-32x32.png	HTTP	120	pfc:True server:web-server-oci provider:oci
11	PFC-HTTP-glez-cloud-wp	https://glez.cloud/wp-content/uploads/2022/05/cropped-bitmoji-cloud.png	HTTP	120	pfc:True server:plesk-axarnet provider:axarnet

[129] Consultar en el documento ASIR2.PFC.7.Services2Monitor.PabloGonzález

12	PFC-HTTP-plesk-glez-cloud	https://plesk.glez.cloud:8443/	HTTP	120	pfc:True server:plesk-axarnet provider:axarnet
13	PFC-HTTP-manage-dns-glez-cloud	https://manage-dns.glez.cloud/	HTTP	120	pfc:True server:ns1-clouding provider:clouding
14	PFC-HTTP-mail-glez-cloud	http://mail.glez.cloud/	HTTP	120	pfc:True server:mail-clouding provider:clouding

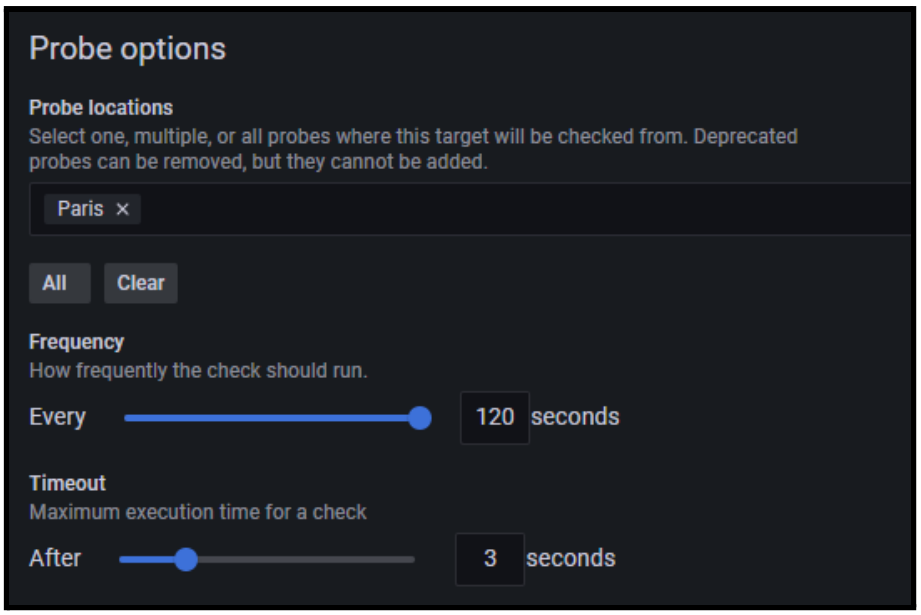
8.3.1. Pasos para creación de check

Vamos a crear el siguiente *Check* utilizando el servicio *Grafana Cloud Synthetic Monitoring*[130]. Seleccionamos tipo DNS, indicamos un nombre para el job, así como el objetivo (el dominio a comprobar).

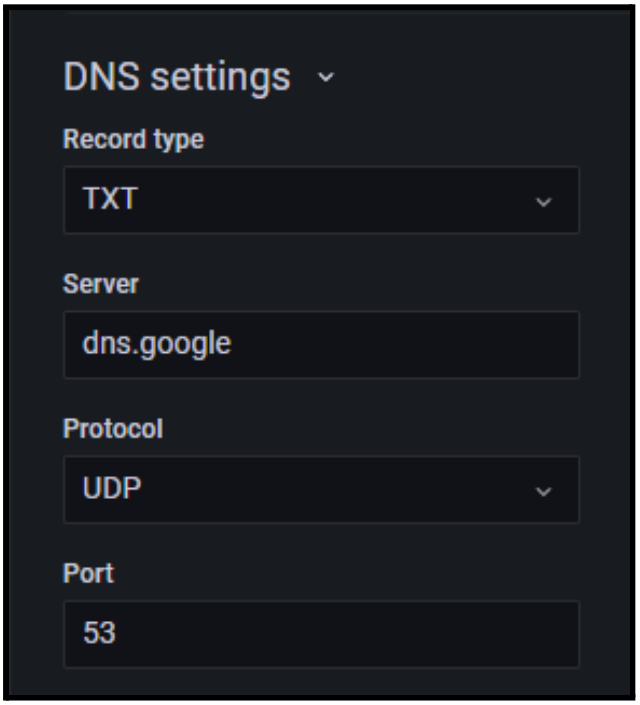


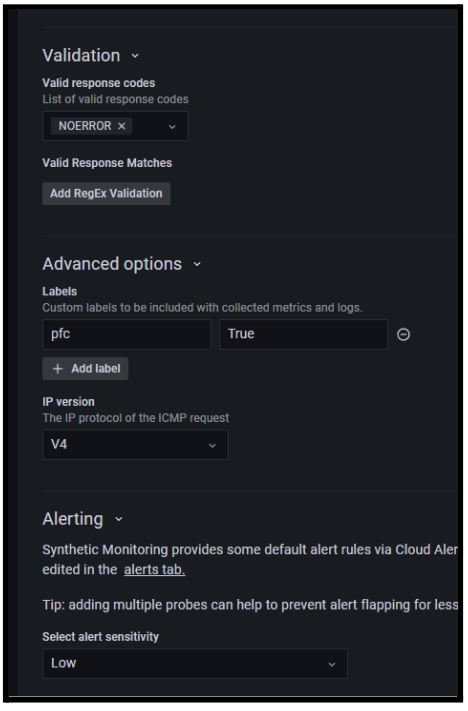
Se comprobará desde París, cada 120 segundos con un timeout de 3 segundos.

[130] https://grafana.com/docs/grafana-cloud/synthetic-monitoring/



Se configura a su vez el tipo de consulta como TXT, al servidor de Google en protocolo UDP. Para la validación, simplemente comprobaremos que no hay error en la respuesta. Asignaremos la etiqueta `pfc:True` para agrupar todos los checks del Trabajo y la sensibilidad la configuraremos en baja para el alertado.

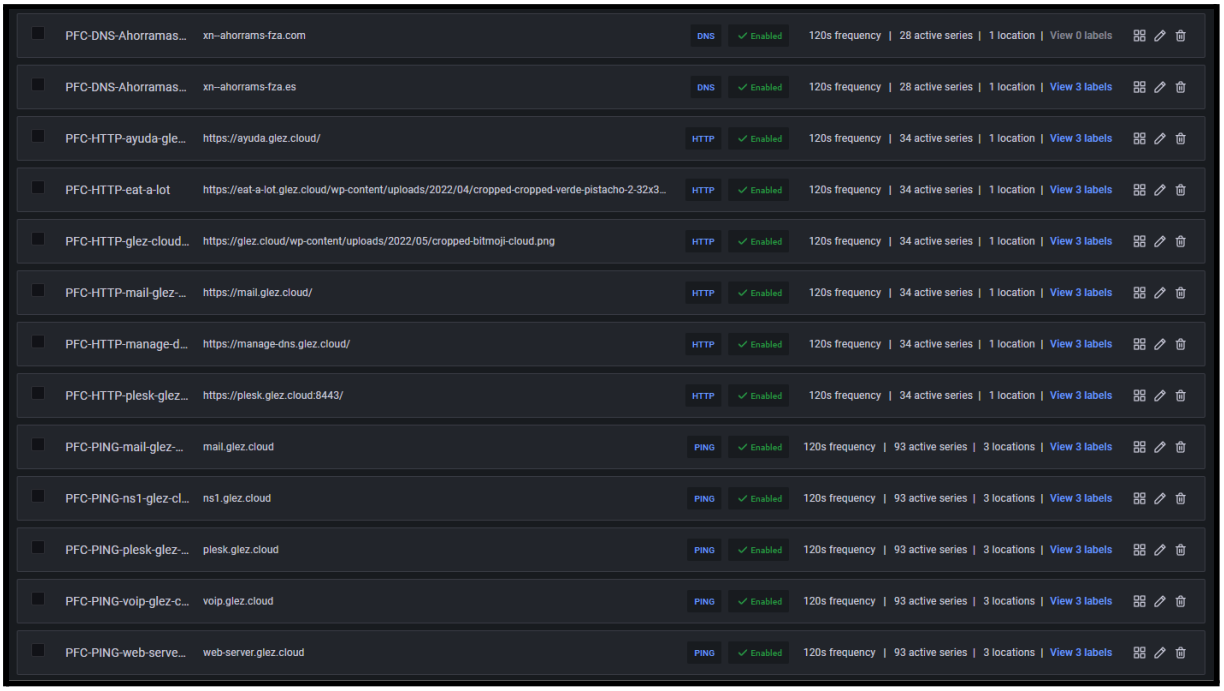




También se han creado etiquetas para el servidor y el proveedor de infrastructura.

8.3.2. Checks creados en Grafana

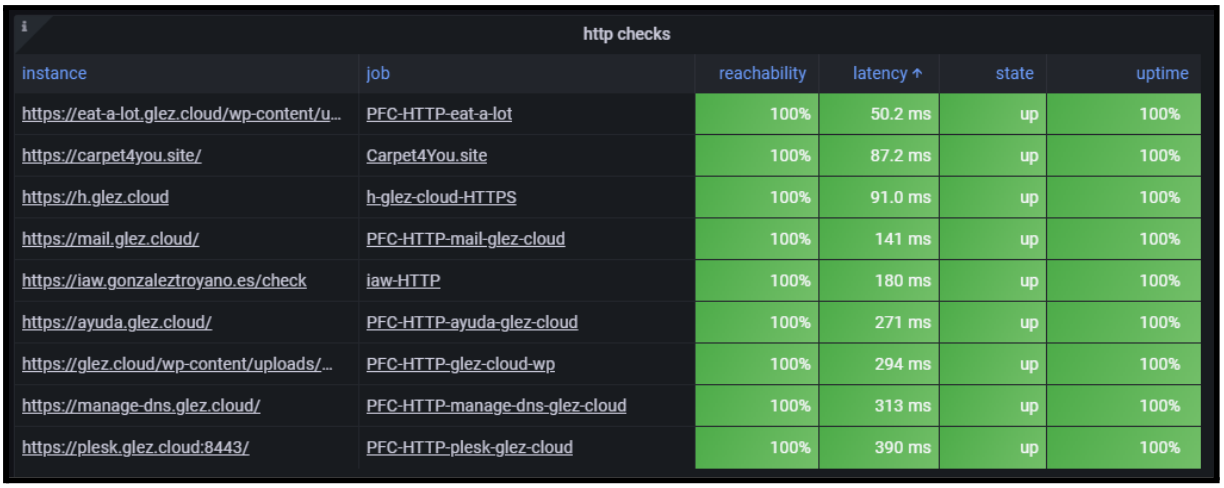
Se crean los siguientes checks en Grafana Synthetic Monitoring. En la imagen se ven desactivados, pero serán activados todos posteriormente.



8.3.3. Ejemplos de *dashboards*

La siguiente captura muestra el panel de resumen de la monitorización de Grafana, con el foco en las pruebas HTTP y ping de Synthetic Monitoring

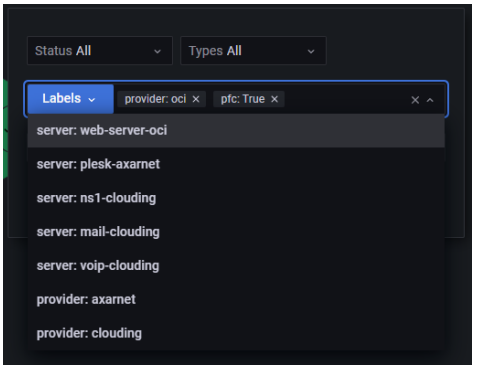
Podemos ver los detalles sobre disponibilidad y latencia desplegando el gráfico:

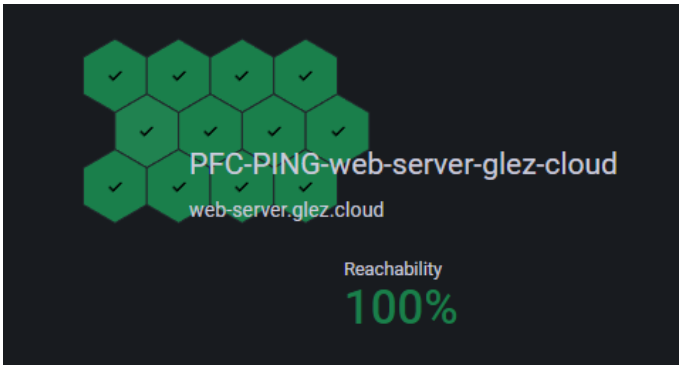


http checks

instance	job	reachability	latency ↑	state	uptime
https://eat-a-lot.glez.cloud/wp-content/u...	PFC-HTTP-eat-a-lot	100%	50.2 ms	up	100%
https://carpet4you.site/	Carpet4You.site	100%	87.2 ms	up	100%
https://h.glez.cloud	h-glez-cloud-HTTPS	100%	91.0 ms	up	100%
https://mail.glez.cloud/	PFC-HTTP-mail-glez-cloud	100%	141 ms	up	100%
https://iaw.gonzaleztroyano.es/check	iaw-HTTP	100%	180 ms	up	100%
https://ayuda.glez.cloud/	PFC-HTTP-ayuda-glez-cloud	100%	271 ms	up	100%
https://glez.cloud/wp-content/uploads/...	PFC-HTTP-glez-cloud-wp	100%	294 ms	up	100%
https://manage-dns.glez.cloud/	PFC-HTTP-manage-dns-glez-cloud	100%	313 ms	up	100%
https://plesk.glez.cloud:8443/	PFC-HTTP-plesk-glez-cloud	100%	390 ms	up	100%

Tenemos la posibilidad de filtrar en base a etiquetas (ver estados de un único proveedor o servidor, por ejemplo). Grafana también incluye una vista para poder ver de un vistazo el estado de los servicios:





También existe la posibilidad de ver los detalles para un único *check*:

8.4. Monitorización de hosts con Grafana

El agente de Grafana recoge y envía a Prometheus (el motor de almacenamiento de series temporales) multitud de datos sobre el host en el que se encuentra instalado. Los datos varían desde la carga de CPU, al uso de la memoria RAM, al tráfico de las diferentes interfaces de red, hasta el número de sockets abiertos.

Veamos un ejemplo de *overview* para el servidor Plesk:

En base a estos registros podemos definir alarmas para, por ejemplo, recibir una notificación si el sistema de archivos está a punto de llegar a su límite de almacenamiento. Este sería un ejemplo de regla para el supuesto indicado:

```
(

  node_filesystem_avail_bytes{job="integrations/node_exporter",fstype!=""} /
node_filesystem_size_bytes{job="integrations/node_exporter",fstype!=""} * 100 < 5

and
  node_filesystem_readonly{job="integrations/node_exporter",fstype!=""} == 0

)
```

8.5. Registro de logs con Grafana

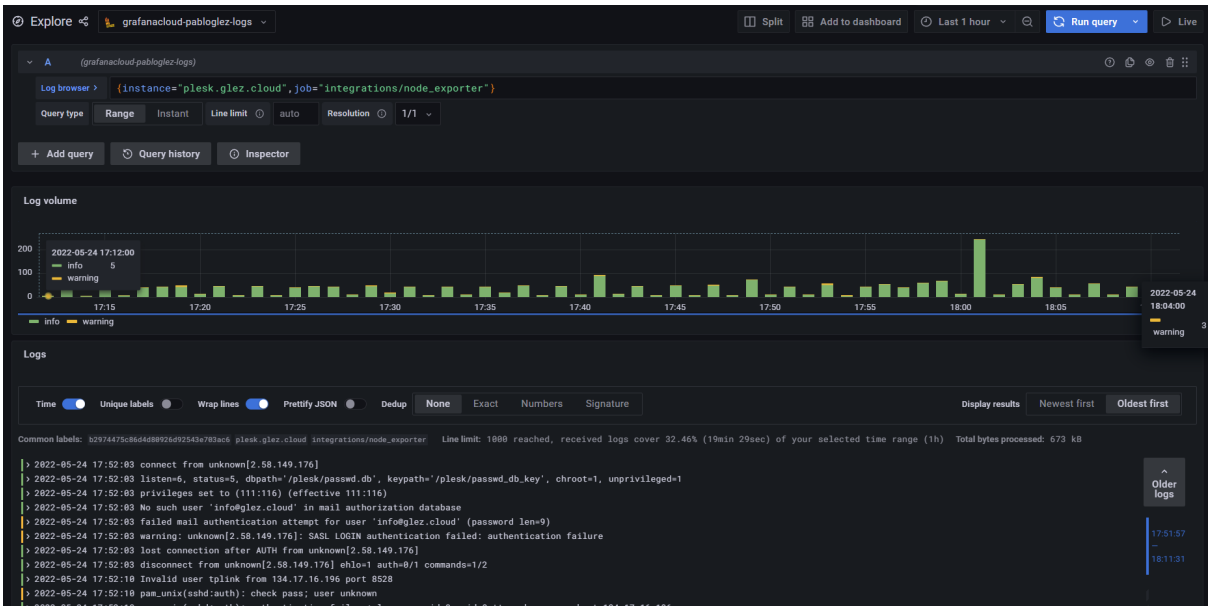
El agente de Grafana también envía a Loki (el motor de almacenamiento de logs) todos los registros del servidor. Por defecto, envía los registros almacenados en `/var/log/`, pero podemos configurar el agente para recolectar registros de cualquier otra ruta del sistema.

Desde la interfaz web de Grafana podemos consultarlos.

Para ver los logs del sistema Plesk podemos ejecutar la siguiente consulta:

```
{instance="plesk.glez.cloud",job="integrations/node_exporter"}
```

Dando como resultado:

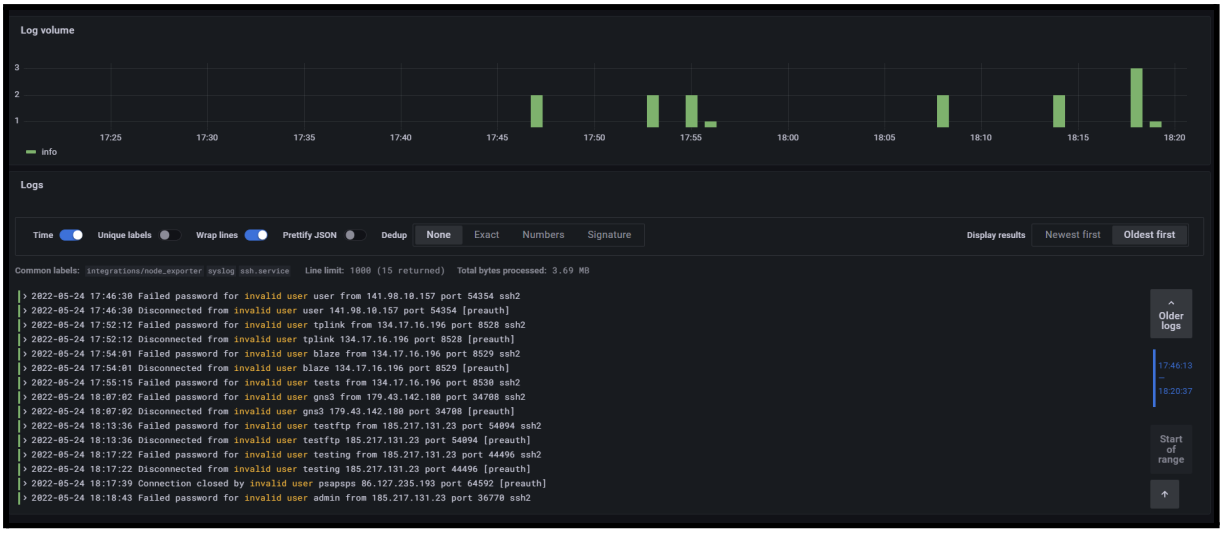


También podríamos hacer una *query* para ver todos los intentos fallidos de inicio de sesión en el sistema, o de intentos de inicio de sesión con usuarios genéricos:

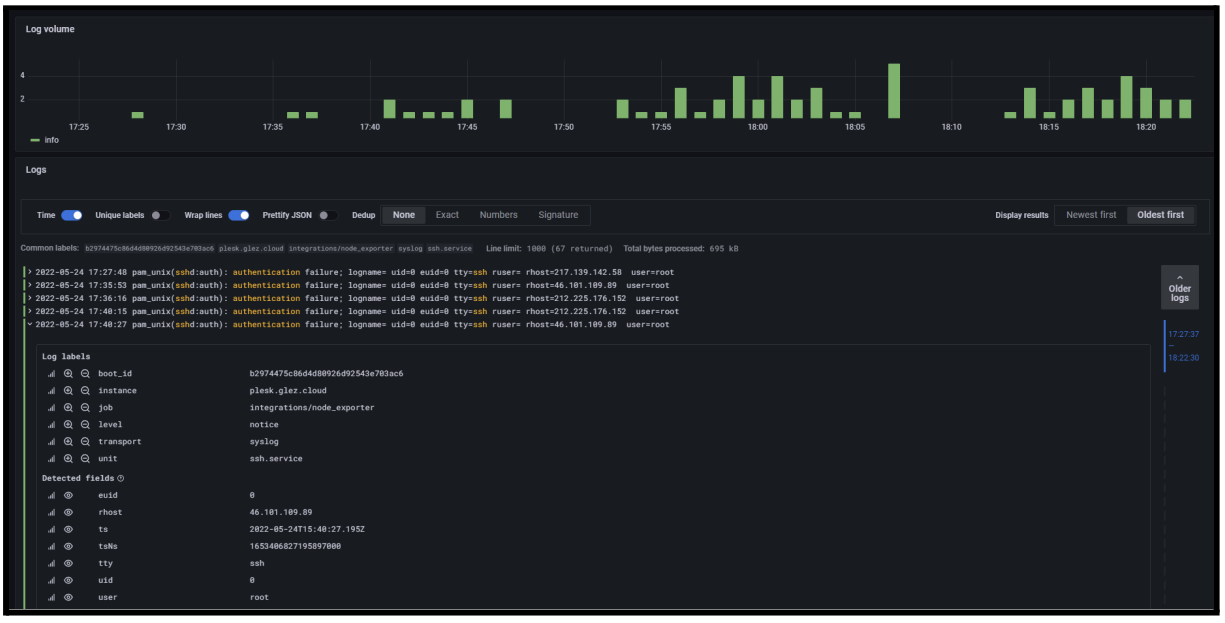
```
{instance="plesk.glez.cloud",job="integrations/node_exporter"}
|= "ssh"|="authentication"
```

```
{instance=~"ns1.glez.cloud|plesk.glez.cloud|voip.glez.cloud|web-server.glez.cloud"}|="invalid user"
```

Dando las consultas superiores los siguientes resultados, respectivamente:



También resulta interesante ver los detalles del registro. Todos los valores aquí mostrados pueden ser usados como filtros (pasando la query de Loki a través del selector “ `| logfmt` ”). También podemos filtrar únicamente por un valor y/o eliminar otros de los resultados.



Los registros se almacenan durante 14 días, aunque este valor es configurable.

En base a registros también podemos definir alarmas (eventos de inicio de sesión, elevación de privilegios, etc)

9. Gestión y acceso a la documentación

Toda la documentación estará disponible en el siguiente enlace:

pfc.gonzaleztroyano.es

En dicha página se pueden encontrar enlaces a este mismo documento (también a los diferentes capítulos del mismo de forma independiente). A su vez, es posible descargar las diapositivas realizadas para la exposición del proyecto desde el siguiente enlace:

pfc.gonzaleztroyano.es/slides

Toda la documentación, trabajo realizado y código desarrollado es accesible de forma pública en el siguiente repositorio de GitHub. A lo largo de todo el documento hay numerosas referencias a este recurso.

github.com/gonzaleztroyano/ASIR2-PFC

Si se accediera a este documento a través de la versión impresa del mismo, puede ser complicado observar con todo detalle algunas imágenes. Como solución a este problema, se ha creado un sitio web en el que introduciendo el número de página y el orden de la imagen en ella (contando como inicial la que se encuentre más cercana a la esquina superior izquierda y en orden tradicional de escritura occidental) se puede ampliar la imagen deseada.

El código se encuentra publicado en el anteriormente mencionado repositorio de GitHub. Se puede acceder a la herramienta desde el siguiente enlace:

pfc.gonzaleztroyano.es/image-viewer

En la siguiente página se puede ver un ejemplo. Nótese que la página debe coincidir con el número visto en la esquina inferior derecha.

También en este sitio web están accesibles una serie de grabaciones en las que se pueden ver los distintos sistemas en funcionamiento. Está categorizado en secciones, una para servicio/objetivo/sección del trabajo. Las categorías son *PowerDNS-Admin*, *Mailcow*, *Script de gestión servidor hosting*, *FreePBX/Asterisk: VoIP*, *Plesk*, *osTicket* y *Grafana/Monitoring*.

El código del vídeo es pasado en la dirección (método GET), leído por un script en JavaScript y consultada la referencia en un JSON. En base a este JSON, se modifica el título de la página, el h2 visible y el vídeo embebido.

Esta lista de vídeos está disponible en la página principal de la web de publicación de documentación de este Proyecto:

pfc.gonzaleztroyano.es

Este sitio está alojado en Firebase. El contenido estático (documentos PDF e imágenes en su mayoría) es servido desde un *bucket* de Google Cloud Storage.

Como prueba de concepto (PoC), se ha generado el siguiente *script* que permite acortar enlaces para este proyecto. Se usa la solución *Dynamics Links*[131] de Firebase. El *script* está desarrollado en Python.

La idea es guardar en un fichero CSV un derivado de la hoja de cálculo que se puede ver a continuación. El *script* recorrerá el CSV referenciado y realizará la correspondiente llamada a la API de Firebase, devolviendo el enlace acortado:

#	Identificador numérico	Título	Identificador extenso - ASIR2 Global	Enlace corto	Enlace completo
0	0	PFC_Final	ASIR2.PFC.0.ProyectoFinalCiclo.PabloGonzález	gonzaleztroyano.es/pfc	https://drive.google.com/open?id=1BoPukCfRb2Fdoqjl
1	1	Anteproyecto_Doc	ASIR2.PFC.1.Anteproyecto.PabloGonzález	https://pglez.es/nAWo	https://drive.google.com/open?id=1vdWbrv-Dqdkfjtd7A
2	1	Anteproyecto_PDF	ASIR2.PFC.1.Anteproyecto.PabloGonzález.pdf	https://pglez.es/vyfY	https://drive.google.com/open?id=1xVJ-hEdceDinAeNJ
3	2	Resources_Doc	ASIR2.PFC.2.Resources.PabloGonzález	https://pglez.es/5y6Q	https://drive.google.com/open?id=1A6citxApX3zq2DM
4	3	Proyect_IDs_Sheet	ASIR2.PFC.3.IdentificadoresDelProyecto.PabloGonzález	https://pglez.es/JSMb	https://drive.google.com/open?id=1-y2cm6lh2msezZ0(

Si bien no ha sido aplicado a todos los enlaces, como PoC ha demostrado su correcto funcionamiento.

El script se puede consultar a continuación:

```
#!/bin/env python3

import requests
import json

def acortar(title, link):
    key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    url = "https://firebasedynamiclinks.googleapis.com/v1/shortLinks?key=" + key
    payload = '{"dynamicLinkInfo":{"domainUriPrefix":"https://[@]pglez[@]es","link":"'+
link +
'","navigationInfo":{"enableForcedRedirect":"True"},"socialMetaTagInfo":{"socialTitle":"
' + title + ' - FCT Pablo González","socialDescription":"Documentación y enlaces del
Proyecto Fin de Ciclo de Pablo González Troyano en 2º
ASIR","socialImageLink":"https://[@]gonzaleztroyano.es[@]content[@]pfc-media[@]png"}},"s
uffix":{"option":"SHORT"}}'
    r = requests.post(url,data=payload)
    print(r.text)
    r_j = json.loads(r.text)
    print(r_j["shortLink"])


fichero = open("./file.csv","r")
for linea in fichero:
    linea = linea.split(',')
    titulo = linea[0]
    enlace = linea[1][:-1]
    acortar(titulo,enlace)
```

[131] https://firebase.google.com/docs/dynamic-links/rest

También como prueba de concepto se han publicado como registros TXT algunos enlaces a los recursos de este proyecto.

Esta forma de publicación surge como reflexión a la expresión “El servicio DNS es una base de datos distribuida”. Es un mantra repetido por muchas personas. Si bien no es del todo cierto puesto que dependemos de servidores raíz y autoritativos, me pareció un tema interesante.

Se han publicado bajo el siguiente subdominio:

```
pfc.gonzaleztroyano.es
```

La nomenclatura es sencilla y se basa en un identificador único para cada documento. Este identificador se añade como subdominio al indicado sobre este texto. Como referencia, en la siguiente tabla se muestran algunos identificadores, aunque hay muchos más (uno para cada documento):

ID	Titulo	Titulo Global en ASIR
0	PFC_Final	ASIR2.PFC.0.ProyectoFinalCiclo.PabloGonzález
1	Anteproyecto_Doc	ASIR2.PFC.1.Anteproyecto.PabloGonzález
1	Anteproyecto_PDF	ASIR2.PFC.1.Anteproyecto.PabloGonzález.pdf
2	Resources_Doc	ASIR2.PFC.2.Resources.PabloGonzález
3	Proyect_IDs_Sheet	ASIR2.PFC.3.IdentificadoresDelProyecto.PabloGonzález

Para consultar los detalles, basta con lanzar una (o varias) consulta DNS:

```
dig TXT {0..3}.pfc.gonzaleztroyano.es +short @1.1.1.1
```

La consulta anterior busca los IDs entre 0 y 1, danto como resultado las siguientes respuestas DNS:

```
"Proyecto_FINAL - https://pfc.gonzaleztroyano.es/PFC-ASIR-PabloGonzalezTroyano.pdf"
"Anteproyecto_Doc - https://drive.google.com/open?id=1vdWbrv-Dgdkfjtd7AJz_F8WA1flJ5wTnFxXu4yt_yd0"
"Anteproyecto_PDF - https://drive.google.com/open?id=1xVJ-hEdceDinAeNzYE_RFcqEjS8qLPAS"
"Resources_Doc - https://drive.google.com/open?id=1A6citxApX3zq2DMc5r1PBcVhbGvEn92ZXDdfHuiJZds"
"Proyect_IDs_Sheet - https://drive.google.com/open?id=1-y2cm6lh2msezZ0GG1ecHgjoYvrfRoOU4tlg5vnC9WM"
```

10. Conclusiones

Obtener una única conclusión del desarrollo de este proyecto es realmente complicado. Podría comenzar por destacar la gran variedad de servicios que se han debido desplegar, configurar y monitorizar. Como usuarios, no somos conscientes de la gran variedad de actores y servicios que usamos para ver una página web o enviar un correo electrónico.

Realizar este proyecto me ha ayudado a comprender de forma más profunda el funcionamiento y la puesta en producción de, por ejemplo, un servidor DNS (y las implicaciones relacionadas como la propagación, DNSSEC y punycode). Ha sido necesario un conocimiento extenso sobre tipos de registro DNS, implicaciones de los distintos TTL, delegación de NameServers de dominios desde los NIC de cada Top-Level Domain, etc.

Simular tratar con clientes reales, y operar en consecuencia, me ha ayudado a enfocar el proyecto no solo desde una parte técnica sino también desde un enfoque comercial y con la intención de facilitar la gestión a los usuarios finales. Por este motivo se han priorizado soluciones que dispusieran de interfaces gráficas de usuario (GUIs). También en la maquetación del sitio en WordPress se han adquirido y aplicado conocimientos de UX y SEO. Así como desarrollo web.

En tanto al desarrollo, se han producido una serie de funciones y/o scripts que conforman un aplicativo completo de gestión de usuarios y servicios de un servidor web. El enfoque en microfunciones ha permitido una mayor limpieza en el código. Respecto al código y a su ciclo de vida, aunque ha sido desarrollado de forma individual, se han usando soluciones de control de versiones y estrategias que permitirían la colaboración de varios programadores (nomenclatura definida para commits, rama de trabajo separada, etc).

Para la administración del servidor web y la confección de los distintos scripts se han aplicado conocimientos de sistemas operativos, manipulación de archivos y bases de datos, entre muchos otros.

Se ha comprendido que la gestión de un servidor de correo electrónico es altamente complicada. Por este motivo existen soluciones de terceros. Algunas en las que se debe instalar y administrar el aplicativo (como es el caso de Mailcow), y otras siendo completamente administradas (como el caso de proveedores de servicios de *hosting* y soluciones de colaboración como Office 365 y Google Workspace). También relativo al correo electrónico, se han aplicado los conocimientos sobre cabeceras de mensajes, funcionamiento del sistema global y seguridad adquiridos durante el curso. Ha sido fundamental comprender las cabeceras completas para solucionar problemas relacionados con marcado como spam y rechazo de mensajes.

Para la comunicación con los clientes, además del correo electrónico se han puesto en marcha dos sistemas: un sistema de telefonía VoIP basado en Asterisk y una plataforma de gestión de tickets basada en osTicket. Ambas son soluciones de software libre, que se han priorizado durante toda la realización del trabajo.

Durante la implementación del sistema de telefonía VoIP se ha comprobado la dificultad de mantener un sistema de tales características; así como de la complicación para conectar el sistema global y obtener un número telefónico. Se ha combinado con osTicket como sistema de soporte basado en tickets, pues es lo común en el mercado. Combinarlo con la gestión de pedidos ha sido interesante para facilitar el seguimiento a los agentes.

También se ha comprobado la facilidad y la rapidez que implica el uso de soluciones como Plesk, que están especialmente preparadas para el enfoque de este proyecto: simular una empresa de alojamiento web. Por una pequeña suscripción, la administración se vuelve sencilla y escalable, dejando tiempo para acciones de otra índole (menos técnicas pero también necesaria para una empresa).

Todo se ha combinado con un sistema de monitorización basado en Grafana. Configurándolo he aprendido la importancia no solo de instalar, sino también de mantener los sistemas informáticos desplegados asegurando su disponibilidad contínua.

11. Propuestas de mejora

Sería negar la evidencia afirmar que en este proyecto no hay mejoras aplicables. Aquí recojo algunas posibilidades de mejora y ampliación para este Proyecto Fin de Ciclo.

Estas mejoras se me han ido ocurriendo durante la realización del mismo, haciendo deporte y en transporte público; entre otros lugares y situaciones. Son en estos contextos donde, en mi opinión, surgen las ideas más disruptivas y con mayor impacto. En esta sección recojo las más destacadas.

Respecto al sistema DNS, si bien PowerDNS dispone de sistemas de replicación primario-secundario similar al que dispone el servidor BIND (basado en *transfers*, *notify* y AXFR) puede no ser completamente eficiente. Gracias a la variedad de *back-ends* de PowerDNS, sería mucho más eficiente usar la replicación nativa de MySQL entre varios servidores. Esto es debido a una mayor consistencia, tasa de transferencia y amplitud que ofrece la replicación nativa respecto al método AXFR.

En este sentido podemos consultar los siguientes recursos: Generic MySQL backend — PowerDNS Authoritative Server documentation[132], PowerDNS Master Slave DNS Replication with MySQL backend[133] y DNS Modes of Operation — PowerDNS Authoritative Server documentation[134].

A su vez, durante todo el desarrollo del proyecto ha sido necesaria la edición de multitud de archivos de configuración para los distintos servicios. Muchos han sido publicados en GitHub (véase el ejemplo del archivo de configuración de Mailcow[135]). Sería, desde mi punto de vista, una buena posibilidad de mejora aplicar el control de versiones a todos los archivos. No podrían ser públicos pues algunos contienen secretos y/o contraseñas.

En tanto a los secretos, podría ser una buena propuesta de mejora el uso de soluciones de gestión de secretos (*Secret Managers*) para aumentar la seguridad de

[132] https://doc.powerdns.com/authoritative/backends/generic-mysql.html
[133] https://www.claudiokuenzler.com/blog/844/powerdns-master-slave-dns-replication-mysql-backend
[134] https://doc.powerdns.com/authoritative/modes-of-operation.html
[135] https://github.com/gonzaleztroyano/ASIR2-PFC/blob/main/2-mail/mailcow.conf.txt

los mismos. La mayoría de proveedores de nube pública (*GCP, OCI, AWS* y *Azure*) lo ofrecen. También existen multitud de soluciones open-source. Una de las más implantadas es *Hashicorp Vault*[136], que permite instalarlo en un servidor propio.

Aplicar el enfoque de IaC (*Infrastructure as Code*, Infraestructura como código) sería también una buena posibilidad de mejora. Si bien el uso de docker-compose es un primer paso, lo ideal sería avanzar más y (de la mano de la implementación de servidores en nubes públicas) usar soluciones como Terraform[137] para el despliegue de entornos y servidores reduciendo la posibilidad de errores humanos, así como la agilidad y continuidad de los servicios.

Durante la selección de aplicaciones/servicios/soluciones se han priorizado no solo soluciones con implantación en la industria y favorablemente de código abierto, sino también que dispusieran de API. Este punto es importante para la interconexión entre sistemas. Fue uno de los motivos por los que se eligieron PowerDNS-Admin y Mailcow. Si bien con la primera hubo problemas al estar todavía en Desarrollo, la API de Mailcow ha demostrado gran potencia. Mediante esta y un simple script podríamos dar de alta un dominio y multitud de cuentas, obteniendo los datos desde un servidor origen (*IdP* o LDAP, por ejemplo) o un archivo CSV.

Respecto a la monitorización, si bien se ha demostrado extensa y confiable gracias a Grafana, en mi opinión sí hubiera sido interesante disponer de una página de estado pública para que los clientes puedan consultar impactos sobre la infraestructura. Si bien la dedicación en tiempo y esfuerzo no sería excesiva, no se ha llevado a cabo para evitar duplicar la monitorización de estado.

Por último, hubiera sido interesante disponer de un sistema de publicación de documentación moderno, más que un PDF. Aunque es entendible que no es posible por legislación/normativa educativa, aquí se muestra un ejemplo[138] del resultado final de la presentación de documentación usando ReadTheDocs. Este formato permite abrir el trabajo a muchas más personas, haciéndolo accesible públicamente.

[136] https://www.vaultproject.io/
[137] https://www.terraform.io/
[138] https://syad.gonzaleztroyano.es/

12. Bibliografía

A continuación se recogen diferentes recursos consultados a lo largo del desarrollo del proyecto.

https://www.cloudflare.com/es-es/products/zero-trust/zero-trust-network-access/

https://www.atlassian.com/es/agile/scrum/sprints

https://www.cloudflare.com/es-es/learning/dns/what-is-dns/

https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml

https://cloud.google.com/vpc/docs/add-remove-network-tags

https://www.portainer.io/

https://datatracker.ietf.org/doc/html/rfc3492/

https://www-static.ripe.net/static/rnd-ui/atlas/media/brochures/RIPE-Atlas-probes-2015_Spanish.pdf

https://labs.ripe.net/author/suzanne_taylor_muzzin/introducing-ripe-atlas-status-checks/

https://doc.powerdns.com/authoritative/settings.htmhttps://github.com/cloudflare/cloudflared

https://github.com/mailcow/mailcow-dockerized

http://www.postfix.org/

https://community.mailcow.email/

https://mailcow.github.io/mailcow-dockerized-docs/manual-guides/u_e-80_to_443/

https://grafana.com/pricing/

https://grafana.com/oss/loki/

https://grafana.com/docs/

https://grafana.com/docs/loki/latest/

https://www.atlassian.com/es/git/tutorials/using-branches

https://wordpress.org/support/article/how-to-install-wordpress/

https://wp-cli.org/es/

https://man7.org/linux/man-pages/man5/sshd_config.5.html

https://developers.sendinblue.com/docs

https://letsencrypt.org/docs/staging-environment/

https://regex101.com/r/cqc8al/1

https://api.cloudflare.com/

https://www.freepbx.org/

https://gist.github.com/kolosek/f0d1952f784f7f164db145497ce155b6

https://wiki.freepbx.org/display/FOP/Installing+FreePBX+14+on+Ubuntu+18.04

https://docs.aws.amazon.com/polly/index.html

https://docs.plesk.com/en-US/obsidian/advanced-administration-guide-linux/about-this-guide.68553/

https://www.cloudflare.com/es-es/learning/security/glossary/what-is-zero-trust/

https://www.atlassian.com/es/agile/project-management/epics

https://www.rfc-editor.org/rfc/rfc1035.html

https://www.powerdns.com/

https://www.isc.org/software/bind

https://docs.docker.com/engine/install/ubuntu/

https://doc.powerdns.com/authoritative/settings.html#loglevel

https://www.lacnic.net/1000/1/lacnic/ripe-atlas-en-latinoamerica-y-caribe

https://diff.wikimedia.org/2014/07/09/how-ripe-atlas-helped-wikipedia-users/

https://github.com/ipinfo/cli

https://doc.powerdns.com/authoritative/modes-of-operation.html#secondary-operation

https://cloud.google.com/compute/docs/tutorials/sending-mail

https://letsencrypt.org/

https://mailcow.github.io/mailcow-dockerized-docs/

https://www.mail-tester.com/

https://grafana.com/oss/prometheus/exporters/node-exporter/

https://grafana.com/oss/prometheus/

https://prometheus.io/docs/introduction/overview/

https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos

https://git-scm.com/docs/git-log

https://devdocs.prestashop.com/1.7/basics/installation/

https://www.debian.org/doc/manuals/securing-debian-manual/chroot-ssh-env.en.html

https://es.sendinblue.com/api/

https://certificate.transparency.dev/

https://api.wordpress.org/secret-key/1.1/salt/

https://doc.prestashop.com/display/PS17/Instalar+PrestaShop

https://github.com/cloudflare/python-cloudflare

https://www.asterisk.org/

https://www.atlantic.net/vps-hosting/how-to-install-asterisk-and-freepbx-on-ubuntu-20-04/

https://docs.plesk.com/en-US/onyx/reseller-guide/understanding-service-plans-and-subscriptions.64607/

grafana.com/oss/prometheus/exporters/node-exporter/

https://docs.osticket.com/en/latest/index.html