



# Institut Puig Castellar

Santa Coloma de Gramenet



## EduNetGuard

*Sistema automatitzat de control d'accés web en entorns educatius*

### **Projecte de desenvolupament**

CFGS Administració de Sistemes Informàtics en Xarxa

Autors: Adam Reggani, Milan Esposito

Grup: ASIX | Curs acadèmic: 2025–2026



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

## RESUM DEL PROJECTE

EduNetGuard és un projecte de desenvolupament que implementa un sistema automatitzat de control d'accés web per a entorns educatius. La temàtica és el filtratge DNS centralitzat en una xarxa d'aula: pfSense actua com a gateway i tallafoc, Pi-hole com a servidor DNS filtrant, MariaDB com a base de dades central, Apache i PHP com a capa web, i un worker Python executat amb cron per aplicar els canvis automàticament. El professorat configura bloquejos per grup, assignatura i franja horària des d'una interfície web, sense accedir directament a les eines tècniques.

La metodologia seguida ha estat incremental. Primer s'ha muntat la infraestructura de xarxa, després s'ha validat el filtratge DNS, s'ha creat la base de dades, s'ha desenvolupat la web i finalment s'ha automatitzat la sincronització amb Pi-hole. Durant el procés s'han corregit problemes reals, com el conflicte de ports, el desbloqueig de regles i l'accés del professorat a la web.

Com a conclusió, s'ha obtingut un sistema funcional en laboratori, modular i defensable, amb millores futures com HTTPS, còpies de seguretat automàtiques i millor control de DNS over HTTPS.

Paraules clau: filtratge DNS, Pi-hole, pfSense, MariaDB, xarxa educativa, automatització, Apache, Python.

## ABSTRACT

EduNetGuard is a development project that implements an automated web access control system for educational environments. The main topic is centralised DNS filtering in a classroom network: pfSense acts as gateway and firewall, Pi-hole as DNS filtering server, MariaDB as central database, Apache and PHP as the web layer, and a Python worker executed by cron to apply changes automatically. Teachers configure content restrictions by group, subject and time slot from a web interface, without directly accessing the technical tools.

The methodology followed has been incremental. First, the network infrastructure was configured, then DNS filtering was tested, the database was created, the web interface was developed, and finally the synchronisation with Pi-hole was automated. During the process, real issues were corrected, such as port conflicts, rule unblocking problems and teacher access to the web interface.

As a conclusion, the result is a functional laboratory system, modular and defendable, with future improvements such as HTTPS, automatic backups and better control of DNS over HTTPS.

Keywords: DNS filtering, Pi-hole, pfSense, MariaDB, educational network, automation, Apache, Python.

# ÍNDEX

[RESUM DEL PROJECTE](#)

[ABSTRACT](#)

[ÍNDEX](#)

## [1. INTRODUCCIÓ](#)

[1.1 Context](#)

[1.2 Justificació](#)

[1.3 Objectius](#)

[1.3.1 Objectiu general](#)

[1.3.2 Objectius específics](#)

[1.4 Estratègia i planificació](#)

[1.5 Metodologia de treball](#)

[1.6 Estudi econòmic i pressupostari](#)

## [2. DESCRIPCIÓ DEL SISTEMA](#)

[2.1 Anàlisi de requisits](#)

[2.1.1 Requisits funcionals](#)

[2.1.2 Requisits no funcionals](#)

[2.2 Comparativa i selecció de tecnologies](#)

[2.2.1 Tecnologies escollides i funcions dins del projecte](#)

[2.3 Arquitectura i components](#)

[2.3.1 Descripció de les màquines virtuals](#)

[2.3.2 Descripció dels serveis principals](#)

[2.4 Funcionalitats del sistema](#)

## [3. IMPLEMENTACIÓ I VALIDACIÓ](#)

[3.1 Infraestructura de xarxa](#)

[3.2 Configuració de pfSense](#)

[3.3 Instal·lació i configuració de Pi-hole](#)

[3.4 Base de dades MariaDB](#)

[3.4.1 Estructura de les taules principals](#)

[3.5 Capa web i backend PHP](#)

[3.5.1 Pantalles HTML](#)

[3.5.2 Fixers del backend PHP](#)

[3.5.3 Disseny de la interfície i flux de navegació](#)

[3.6 Worker Python i automatització amb cron](#)

[3.6.1 Lògica d'execució del worker](#)

[3.6.2 Automatització amb cron](#)

[3.6.3 Script seed\\_pihole\\_categories.py](#)

[3.7 Panell administrador](#)

[3.8 Seguretat del sistema](#)

[3.9 Problemes trobats i solucions aplicades](#)

[3.9.1 Disputa de ports entre Apache i Pi-hole](#)

[3.9.2 El client alumne no utilitzava Pi-hole com a DNS](#)

[3.9.3 Desbloqueig poc fiable amb regles regex](#)

[3.9.4 Error pel mòdul php-mbstring](#)

[3.9.5 Confusió entre franges configurades i franges actives](#)

[3.9.6 El professorat no podia accedir a srv-filter](#)

[3.9.7 Error amb anadido\\_por i añadido\\_por](#)

[3.9.8 Reset States inestable per SSH](#)

[3.9.9 Errors inicials d'instal·lació de pfSense](#)

[3.9.10 Eliminació d'usuaris amb historial](#)

[3.10 Proves i validació](#)

[3.11 Limitacions i millores futures](#)

[3.12 Relació transversal amb el cicle](#)

#### [4. CONCLUSIONS](#)

[4.1 Conclusions generals](#)

[4.2 Consecució dels objectius](#)

[4.3 Valoració de la metodologia i planificació](#)

[4.4 Visió de futur](#)

#### [5. GLOSSARI](#)

#### [6. BIBLIOGRAFIA](#)

#### [7. ANNEXOS](#)

[Annex A. Màquines virtuals del laboratori](#)

[Annex B. Xarxes, adreces IP i ports](#)

[Annex C. Rutes principals del projecte](#)

[Annex D. Evidències de validació del sistema](#)

[Annex E. Manual breu d'ús](#)

[Annex F. Estat honest del projecte](#)

[Annex G. Categories i dominis de bloqueig](#)

[Annex H. Relació entre objectius, proves i estat](#)

[Nota sobre l'ús d'intel·ligència artificial](#)

# 1. INTRODUCCIÓ

La idea va sorgir d'un problema real: durant les classes, els alumnes tenien accés a tot. Xarxes socials, YouTube, jocs en línia, eines d'IA. No hi havia cap forma fàcil de dir "ara no" sense anar equip per equip. EduNetGuard és la solució que vam decidir construir per resoldre exactament això.

Aquesta memòria explica com ho hem fet, en quin ordre, per quin motiu hem pres cada decisió i quins problemes ens hem trobat pel camí. No tot va sortir com teníem previst, i això també queda reflectit aquí.

## 1.1 Context

L'ús d'Internet en centres educatius és imprescindible. Les classes actuals depenen de plataformes digitals, repositoris, eines de comunicació i serveis web. En un cicle d'informàtica aquesta dependència és especialment clara, perquè moltes pràctiques requereixen màquines connectades a la xarxa.

Tot i això, aquesta connectivitat pot generar problemes durant determinades activitats. En alguns moments pot ser necessari limitar l'accés a xarxes socials, jocs, plataformes de vídeo, serveis de descàrrega o eines d'intel·ligència artificial. Aquests recursos no són intrínsecament negatius, però poden no ser adequats durant una prova, una explicació o una pràctica concreta.

A l'Institut Puig Castellar, com a la majoria de centres, no hi ha cap sistema que permeti a un professor dir "durant la meua classe de base de dades no vull que els alumnes tinguin accés a YouTube, però el professor de la classe d'al costat sí que el pot necessitar". O ho bloqueges per a tothom o no ho bloqueges per a ningú. EduNetGuard neix exactament per resoldre això.

## 1.2 Justificació

La justificació del projecte parteix d'un problema habitual: controlar la navegació d'una aula sense haver de modificar manualment cada equip. Fer canvis client per client no és escalable, genera errors i depèn dels permisos locals de cada màquina.

Una opció alternativa seria bloquejar directament des del tallafoç per IP, però molts serveis web utilitzen CDNs, adreces IP variables o infraestructures distribuïdes, cosa que dificulta aquest enfocament. Un proxy web és una altra alternativa vàlida, però afegeix complexitat i pot donar problemes amb el trànsit HTTPS en un laboratori educatiu.

Vam descartar bloquejar per IP perquè Instagram sol canviar d'adreces i en cinc minuts de proves ja vam veure que no funcionava. Vam descartar un proxy perquè en un laboratori virtual afegir un servei més entre el client i Internet era un problema que no volíem. El filtratge DNS amb Pi-hole era l'opció que tocava menys coses, funcionava sense tocar cap client i es podia automatitzar. Per això el vam escollir.

## 1.3 Objectius

### 1.3.1 Objectiu general

Dissenyar i implementar un sistema automatitzat de control d'accés web per a entorns educatius, basat en filtratge DNS, gestió web per part del professorat, xarxes diferenciades i aplicació dinàmica de bloquejos segons grup, assignatura i franja horària.

### 1.3.2 Objectius específics

1. Crear un laboratori virtual funcional amb pfSense, Ubuntu Server i Ubuntu Desktop sobre VirtualBox.
2. Configurar pfSense com a gateway, tallafoc, servidor DHCP, NAT i punt de control DNS.
3. Implementar LAN\_EDU com a xarxa d'alumnes filtrada per Pi-hole.
4. Implementar LAN\_PROFES com a xarxa independent de professorat, sense bloqueig DNS.
5. Instal·lar i configurar Pi-hole com a servidor DNS filtrant integrat amb el sistema.
6. Dissenyar una base de dades relacional per emmagatzemar professors, assignatures, permisos, horaris, bloquejos i estats.
7. Desenvolupar una interfície web amb login, selecció de grup, horari general, horari personal i gestió de bloquejos.
8. Implementar un panell administrador per gestionar usuaris, contrasenyes, estat de comptes i permisos per assignatura.
9. Desenvolupar un backend PHP que connecti la web amb MariaDB i controli sessions, permisos i persistència.
10. Implementar un worker Python que sincronitza MariaDB amb Pi-hole de forma automàtica.
11. Automatitzar l'execució del worker cada minut amb cron.
12. Diferenciar visualment franges configurades de franges actives a la interfície web.
13. Permetre categories de bloqueig, dominis extra bloquejats i dominis permesos com a excepció.
14. Documentar les decisions tècniques, els problemes trobats, les proves realitzades i les millores futures.

## 1.4 Estratègia i planificació

Des del principi vam tenir clar que no volíem reinventar la roda. Pi-hole ja filtra DNS, pfSense ja és un firewall complet. El que faltava era una capa que els connectés i permetés al professor gestionar-ho sense saber res de xarxes. Aquí és on hem posat la majoria de la feina.

La planificació ha estat incremental. Algunes millores, com LAN\_PROFES, el panell administrador, l'horari personal o la decisió d'usar el camp enabled a Pi-hole, van aparèixer com a resposta a problemes reals detectats durant el desenvolupament.

Fase	Descripció	Hores est.	Resultat esperat	Estat
1	Creació del laboratori virtual	3h	pfSense, servidor i client operatius	Assolit
2	Configuració de xarxa base	4 h	LAN_EDU amb sortida a Internet i DHCP	Assolit
3	Instal·lació i proves de Pi-hole	8 h	Bloquejos DNS validats manualment	Assolit
4	Integració pfSense-Pi-hole	16 h	DNS lliurat per DHCP i redirecció DNS cap a Pi-hole	Assolit
5	Disseny de la base de dades	18 h	Model relacional complet per a usuaris, horaris i regles	Assolit
6	Desenvolupament web i backend	21 h	Login, horaris, gestió de bloquejos i persistència a MariaDB	Assolit
7	Worker Python i cron	24 h	Sincronització automàtica entre MariaDB i Pi-hole	Assolit
8	Xarxa de professorat (LAN_PROFES)	14 h	Xarxa independent sense filtratge DNS i amb accés a la web	Assolit
9	Panell administrador	12 h	Gestió d'usuaris, estat de comptes i permisos per assignatura	Assolit
10	Proves, correccions i documentació	10 h	Sistema validat i memòria tancada	Assolit

Font: elaboració pròpia. Total estimat: 130 hores de treball.

## 1.5 Metodologia de treball

La metodologia aplicada ha estat incremental i iterativa. El sistema s'ha desenvolupat per blocs funcionals: primer la infraestructura de xarxa, després el filtratge DNS amb Pi-hole, a continuació la base de dades, posteriorment la web i el backend, i finalment l'automatització amb Python i cron. S'ha escollit aquesta metodologia perquè els components del projecte estan fortament relacionats entre si: una decisió de xarxa podia afectar el funcionament de la web, i una decisió de base de dades podia afectar el worker. Un enfocament incremental permet validar cada fase abans de construir la següent, i corregir errors quan el cost de fer-ho és baix.

Per al seguiment de la planificació s'ha utilitzat una taula de fases amb hores estimades i estat per a cada bloc de treball, complementada amb un diagrama de Gantt i un full de seguiment setmanal proporcionat pel tutor del projecte. Aquesta combinació ha permès tenir visibilitat sobre l'avanç real sense necessitat d'eines de gestió externes. La revisió de l'estat s'ha fet de forma contínua durant les sessions de treball, comprovant que cada fase funcionés correctament abans de passar a la següent.

Durant les proves van aparèixer imprevistos que van obligar a canviar algunes decisions: el conflicte de ports entre Apache i Pi-hole, la manca del mòdul php-mbstring, els problemes de desbloqueig a Pi-hole o el retorn de trànsit entre xarxes de professorat i alumnes. Tots aquests imprevistos estan documentats a la secció 3.9 amb la causa identificada i la solució aplicada.

## 1.6 Estudi econòmic i pressupostari

El projecte s'ha desenvolupat íntegrament amb programari lliure o gratuït, cosa que redueix el cost directe a zero en un entorn de laboratori sobre VirtualBox. En una implantació real s'hauria de considerar el cost del maquinari, el temps de configuració i el manteniment posterior.

Element	Cost estimat	Observacions
Oracle VirtualBox	0 €	Virtualització del laboratori
pfSense CE	0 €	Firewall i gateway
Ubuntu Server	0 €	Servidor principal
Ubuntu Desktop	0 €	Client de proves
Pi-hole	0 €	Filtratge DNS
MariaDB	0 €	Base de dades
Apache + PHP	0 €	Servidor web i backend
Python + cron	0 €	Automatització
Maquinari virtualitzat	0 €	S'aprofita l'equip amfitrió existent
Mini PC o servidor real (desplegament físic)	150–400 €	Opció per a un centre real
Temps de treball (~153 h)	Cost principal	Instal·lació, proves, depuració i documentació

Font: elaboració pròpia.

El cost de manteniment en un centre educatiu real dependria principalment de la revisió de logs, còpies de seguretat, actualitzacions de sistema i gestió d'usuaris. La solució podria executar-se en un servidor existent o en un mini PC de baix consum.

## 2. DESCRIPCIÓ DEL SISTEMA

Aquest capítol presenta el sistema EduNetGuard des d'un punt de vista conceptual i de disseny: quins requisits ha de complir, quines tecnologies s'han escollit i per quin motiu, com s'organitzen els components i quines funcionalitats ofereix al professorat i a l'administrador.

### 2.1 Anàlisi de requisits

#### 2.1.1 Requisits funcionals

Requisit funcional	Descripció	Estat
Login i sessió	Un professor pot iniciar i tancar sessió amb usuari i contrasenya. El backend comprova que el compte estigui actiu.	Complet
Diferenciació de rols	El sistema diferencia entre professor normal i administrador. Cada rol veu i pot fer coses diferents.	Complet
Selecció de grup	El professor pot seleccionar ASIX_A o ASIX_B i accedir a l'horari corresponent.	Complet
Horari personal	El professor veu només les assignatures que té assignades, evitant que modifiqui franges alienes.	Complet
Bloqueig categories per	El sistema permet activar categories de contingut (IA, Jocs, Xarxes socials, etc.) en una franja.	Complet
Dominis bloquejats extra	El professor pot afegir dominis concrets que no pertanyen a cap categoria predefinida.	Complet
Excepcions permeses	El professor pot declarar dominis permesos que prevalen sobre qualsevol bloqueig actiu.	Complet
Persistència MariaDB a	Tota la configuració es guarda a MariaDB, no directament a Pi-hole.	Complet
Aplicació automàtica	El worker llegeix MariaDB i aplica els bloquejos a Pi-hole sense intervenció manual.	Complet
Desactivació horari per	Quan acaba una franja, el worker desactiva automàticament els bloquejos corresponents.	Complet
Visualització configurada/actiu	La web diferencia franges que tenen configuració guardada de les que estan actives en aquell moment.	Complet
Panell administrador	L'administrador pot crear, modificar, desactivar i eliminar usuaris, i assignar permisos per assignatura.	Complet
Xarxa de professorat	El professorat disposa d'una xarxa pròpia que no queda afectada pels bloquejos aplicats als alumnes.	Complet

Font: elaboració pròpia.

## 2.1.2 Requisits no funcionals

Requisit funcional	no	Descripció
Modularitat		El sistema separa xarxa, DNS, base de dades, backend, web i automatització en components independents.
Mantenibilitat		El sistema és fàcil de depurar gràcies als logs del worker, d'Apache i del sistema.
Usabilitat		La interfície web és usable per professorat sense coneixements avançats de xarxes o sistemes.
Consistència dades	de	La base de dades evita duplicats i manté una estructura coherent entre professors, permisos i regles.
Escalabilitat		El sistema es pot ampliar amb noves classes, assignatures, professors o categories sense canviar l'arquitectura.
Seguretat contrasenyes	de	Les contrasenyes noves o modificades es guarden amb password_hash de PHP (bcrypt).
Tolerància a funcions opcionals		El sistema funciona correctament sense Reset States de pfSense, que queda com a millora opcional.
Documentació honesta		La memòria recull les limitacions reals, les millores futures i l'estat exacte de cada component.

Font: elaboració pròpia.

## 2.2 Comparativa i selecció de tecnologies

La tria de tecnologies s'ha fet valorant alternatives reals. No es buscava només bloquejar webs, sinó fer-ho de manera centralitzada, gestionable pel professorat i sense modificar cada equip client.

Comparació	Alternativa valorada	Decisió final i motiu
Pi-hole vs AdGuard Home	AdGuard Home és igualment potent i té una interfície moderna.	Es manté Pi-hole perquè la integració directa amb gravity.db permet activar i desactivar entrades de forma controlada amb el camp enabled, cosa que encaixa millor amb el disseny del worker.
pfSense vs OPNsense	OPNsense té una interfície més moderna i un cicle d'actualitzacions actiu.	Es manté pfSense perquè ja resol gateway, DHCP, NAT, DNS redirect i separació de xarxes. Canviar no aportava avantatge per al laboratori.
Apache vs Nginx	Nginx és més lleuger i ràpid en entorns de producció.	Es manté Apache per la seva integració directa amb PHP i per facilitar el desplegament. El conflicte amb Pi-hole al port 80 es resol movent EduNetGuard al port 8080.
MariaDB vs PostgreSQL	PostgreSQL és un SGBD relacional molt complet i robust.	Es manté MariaDB perquè cobreix les necessitats del projecte i la integració amb PHP via PDO és directa i senzilla.
cron vs systemd timer	systemd timer ofereix més integració amb el sistema i registre de logs.	Es manté cron perquè l'execució periòdica del worker cada minut és una tasca simple que no requereix la complexitat addicional de systemd timer.
PHP vs Flask (Python)	Flask permetria un backend Python unificat amb el worker.	Es manté PHP per la seva integració natural amb Apache i perquè el backend ja estava funcionant. Afegir Flask hauria requerit un servei addicional.

Font: elaboració pròpia.

## 2.2.1 Tecnologies escollides i funcions dins del projecte

Tecnologia	Versió / distribució	Funció dins del projecte
pfSense CE	2.8.1	Gateway, firewall, DHCP, NAT, DNS redirect i separació de xarxes
Pi-hole	Versió instal·lada al laboratori	Servidor DNS filtrant; gestiona bloquejos via gravity.db
Ubuntu Server	22.04 LTS	Servidor principal que allotja tots els serveis interns
Ubuntu Desktop	22.04 LTS	Client de proves per simular alumne i professor
MariaDB	Versió instal·lada al laboratori	Base de dades relacional central del projecte
Apache2	2.4	Servidor web que serveix EduNetGuard al port 8080
PHP	8.x	Backend: sessions, permisos, lògica de guardat i panell admin
HTML / CSS / JS	—	Interfície web per al professorat i l'administrador
Python 3 + PyMySQL	3.x	Worker que sincronitza MariaDB amb Pi-hole automàticament
Cron	—	Executa el worker cada minut com a root
SQLite3	—	Motor intern de Pi-hole per a gravity.db
Oracle VirtualBox	7.x	Virtualitzador del laboratori

Font: elaboració pròpia.

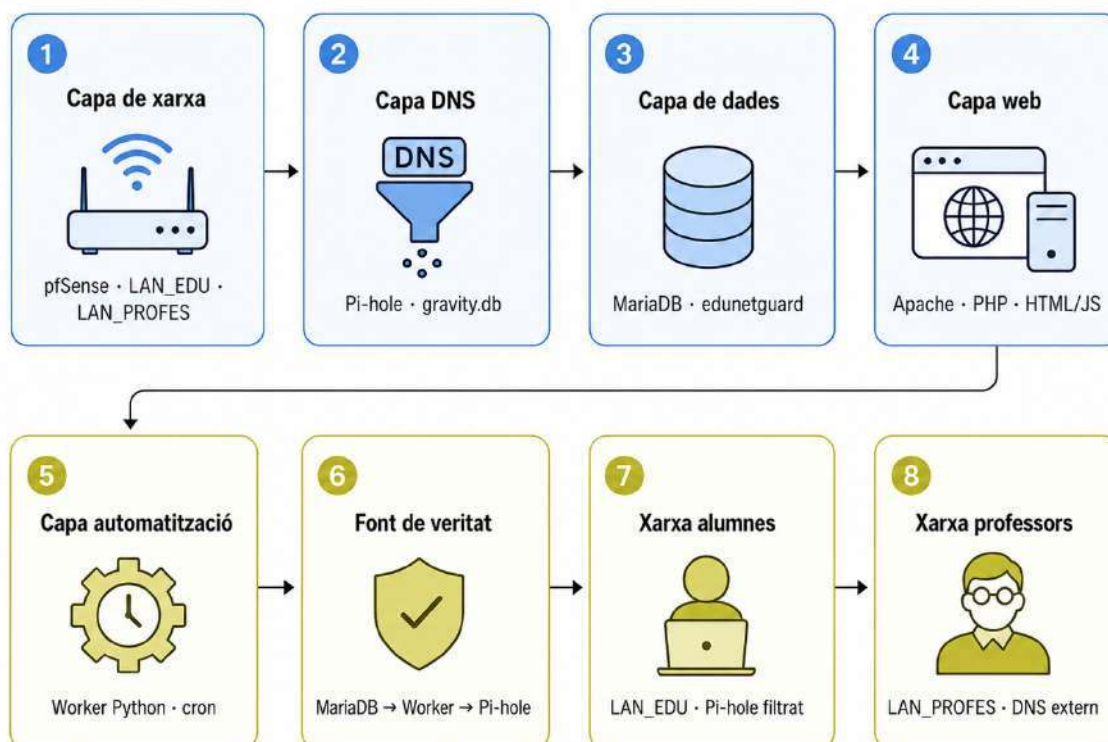
## 2.3 Arquitectura i components

El sistema té sis capes ben separades. Cadascuna fa una cosa i no toca les altres. Això ens va ajudar molt quan hi havia errors: si el bloqueig no s'aplicava, sabíem que el problema era al worker o a Pi-hole, no a la web. Si el login no funcionava, era PHP i MariaDB, no cron.

**Les capes del sistema, de baix a dalt, són:**

- Capa de xarxa: pfSense, LAN\_EDU (10.0.0.0/24) i LAN\_PROFES (10.0.1.0/24).
- Capa DNS: Pi-hole i la base interna gravity.db (SQLite).
- Capa de dades: MariaDB i la base de dades edunetguard.
- Capa web i backend: Apache, PHP i fitxers HTML/CSS/JavaScript.
- Capa d'automatització: worker Python i cron.
- Capa client: màquina Ubuntu Desktop usada com a alumne o professor.

**El flux principal del sistema és:**



*Figura 1. Arquitectura general d'EduNetGuard. Font: elaboració pròpia amb suport d'eines d'intel·ligència artificial per a la generació visual.*

La decisió més important de l'arquitectura és que MariaDB actua com a font de veritat. Això vol dir que la configuració educativa (qui ha bloquejat què, en quina franja i per a quin grup) es guarda a la base de dades. Pi-hole només aplica el bloqueig DNS final i no coneix horaris, professors ni grups. El worker és l'únic component que llegeix MariaDB i modifica Pi-hole, cosa que evita que la web manipuli directament serveis crítics de xarxa.

### 2.3.1 Descripció de les màquines virtuals

Màquina	Sistema	IP / Xarxa	Rol al projecte
fw-edunetguard	pfSense CE 2.8.1	WAN: 10.0.2.15 LAN: 10.0.0.1 PROFES: 10.0.1.1	Gateway, tallafoc, DHCP, NAT, DNS redirect i segmentació de xarxes
srv-filter	Ubuntu Server 22.04	LAN: 10.0.0.51 Pont: 192.168.254.83	Servidor principal: allotja Pi-hole, MariaDB, Apache, PHP, worker Python i cron
Client de proves	Ubuntu Desktop 22.04	Alumne: 10.0.0.50 Professor: 10.0.1.50	Simula alumne (LAN_EDU) i professor (LAN_PROFES) activant una interfície o l'altra

Font: elaboració pròpia.

### 2.3.2 Descripció dels serveis principals

De les cinc eines principals, pfSense és la que menys toca el codi però la que més afecta tot. Si pfSense no funciona bé, no hi ha xarxa, no hi ha DNS i no hi ha web. La seva configuració detallada es descriu a la secció 3.2.

Pi-hole és el que realment aplica els bloquejos als alumnes, però el professorat no l'obre mai directament: tots els canvis arriben a través del worker. El mecanisme de bloqueig es descriu a la secció 3.3.

MariaDB és on viu tota la lògica educativa: professors, assignatures, permisos, horaris i regles. L'estructura de taules es descriu a la secció 3.4.

Apache i PHP serveixen la web al port 8080 per no xocar amb Pi-hole, que conserva el port 80. La capa web es descriu a la secció 3.5.

El worker Python és el que ho connecta tot: llegeix MariaDB i actualitza Pi-hole cada minut via cron. El seu funcionament intern es descriu a la secció 3.6.

## 2.4 Funcionalitats del sistema

Funcionalitat	Descripció
Autenticació	Login i logout amb validació de compte actiu i contrasenya hashejada (bcrypt). Sessió PHP per identificar el rol.
Selecció de grup	Permet triar entre ASIX_A i ASIX_B per accedir a l'horari corresponent.
Horari general	Mostra totes les franges del grup seleccionat i permet navegar cap a la gestió de bloquejos.
Horari personal	Mostra únicament les assignatures assignades al professor que ha iniciat sessió. Evita que modifiqui franges d'altres.
Gestió de bloquejos	Permet activar categories, afegir dominis extra bloquejats i definir excepcions permeses per a una franja concreta.
Diferenciació configurada/actiu	La web mostra clarament si una franja té configuració guardada però no és activa, o si és activa en aquell moment.
Panell administrador	Gestió completa d'usuaris: crear, editar, canviar contrasenya, activar/desactivar i eliminar. Assignació de permisos per assignatura.
Sincronització automàtica	El worker s'executa cada minut via cron. Aplica o desactiva bloquejos sense que el professor hagi de fer res.
Xarxa de professorat	LAN_PROFES permet al professorat treballar i accedir a la web sense quedar afectat pels bloquejos dels alumnes.

Font: elaboració pròpia.

### 3. IMPLEMENTACIÓ I VALIDACIÓ

Aquest capítol explica com s'ha construït EduNetGuard, en quin ordre, quines decisions s'han pres sobre la marxa i per quin motiu. No és una repetició del capítol anterior: aquí s'entra en el detall tècnic de la configuració real, els problemes trobats i les proves realitzades per validar el sistema.

#### 3.1 Infraestructura de xarxa

El laboratori s'ha muntat sobre Oracle VirtualBox. pfSense actua com a router i tallafoc central amb tres interfícies: WAN per NAT cap a Internet, LAN\_EDU per als alumnes i LAN\_PROFES per al professorat.

Xarxa	Rang	Gateway	DNS assignat	Funció
LAN_EDU	10.0.0.0/24	10.0.0.1	10.0.0.51 (Pi-hole, forçat)	Xarxa d'alumnes amb filtratge DNS actiu
LAN_PROFES	10.0.1.0/24	10.0.1.1	1.1.1.1 / 8.8.8.8 (extern)	Xarxa de professorat sense bloqueig DNS
WAN pfSense	10.0.2.15/24	VirtualBox NAT	Propi de l'entorn	Sortida a Internet de tot el laboratori

Font: elaboració pròpia.

A LAN\_EDU, pfSense lliura 10.0.0.51 com a DNS via DHCP i força les consultes del port 53 cap a Pi-hole. Això evita que un alumne pugui canviar manualment el DNS per saltar-se el filtratge DNS clàssic.

A LAN\_PROFES, el professorat utilitza DNS extern (1.1.1.1 i 8.8.8.8) i no queda filtrat per Pi-hole. El problema que va aparèixer és que els professors necessitaven accedir a la web EduNetGuard allotjada a 10.0.0.51, que pertany a LAN\_EDU. El trànsit de retorn des de srv-filter no trobava la ruta correcta cap a 10.0.1.0/24. La solució va ser afegir una regla NAT Outbound a pfSense que emmascara l'origen de les connexions provinents de LAN\_PROFES quan van cap a 10.0.0.51.

Els paràmetres exactes de la regla NAT Outbound aplicada es detallen a l'Annex B.

Aquesta solució es va escollir perquè no requeria modificar les rutes de srv-filter ni afegir rutes estàtiques al servidor, i mantenia el control de xarxa centralitzat a pfSense.

## 3.2 Configuració de pfSense

La configuració de pfSense s'ha fet íntegrament via el WebConfigurator (port 443). Les funcions configurades han estat:

- Interfícies: WAN (NAT VirtualBox), LAN (em1, 10.0.0.1/24) i PROFES (em2/OPT1, 10.0.1.1/24).
- DHCP per LAN\_EDU i LAN\_PROFES, amb DNS corresponents a cada xarxa.
- NAT Port Forward al port 53: tot el trànsit DNS del port 53 de LAN\_EDU es redirigeix a 10.0.0.51 (Pi-hole), independentment del DNS que hagi configurat manualment el client.
- Regla de tallafoc a PROFES: pas de tot el trànsit de la subxarxa 10.0.1.0/24 cap a qualsevol destinació.
- NAT Outbound per permetre a LAN\_PROFES accedir a srv-filter (descriu a 3.1).

Durant la configuració inicial es van produir errors d'instal·lació relacionats amb l'arquitectura de la VM, problemes de boot amb ZFS i la necessitat de triar pfSense CE en lloc de pfSense Plus. Tots es van resoldre ajustant els paràmetres de VirtualBox.

## 3.3 Instal·lació i configuració de Pi-hole

Pi-hole s'ha instal·lat a srv-filter (10.0.0.51) i conserva el port 80 per al seu panell web. Apache serveix EduNetGuard al port 8080 per evitar el conflicte entre els dos serveis.

El mecanisme de bloqueig escollit és el camp enabled de la taula domain list dins de gravity.db, la base de dades SQLite interna de Pi-hole. En lloc de crear i eliminar regles contínuament, es va optar per sembrar les categories EDUNET una sola vegada amb el script seed\_pihole\_categories.py, deixant totes les entrades amb enabled=0. Després, el worker les activa amb enabled=1 o les desactiva amb enabled=0 segons les franges actives guardades a MariaDB.

Aquesta decisió es va prendre perquè durant les proves l'enfocament inicial de crear i eliminar regles generava problemes. Per una banda, podien aparèixer entrades duplicades a gravity.db. Per altra banda, el desbloqueig no sempre era fiable, ja que algunes ordres de Pi-hole podien interpretar mal els paràmetres d'eliminació. Amb el sistema basat en enabled=1/0 no cal eliminar entrades, només canviar-ne l'estat.

Les categories sembrades a Pi-hole són IA, Jocs, Xarxes socials, Streaming, Descàrregues, Compres en línia i Fòrums i oci. La llista completa de categories, identificadors EDUNET i dominis associats es recull a l'Annex G.

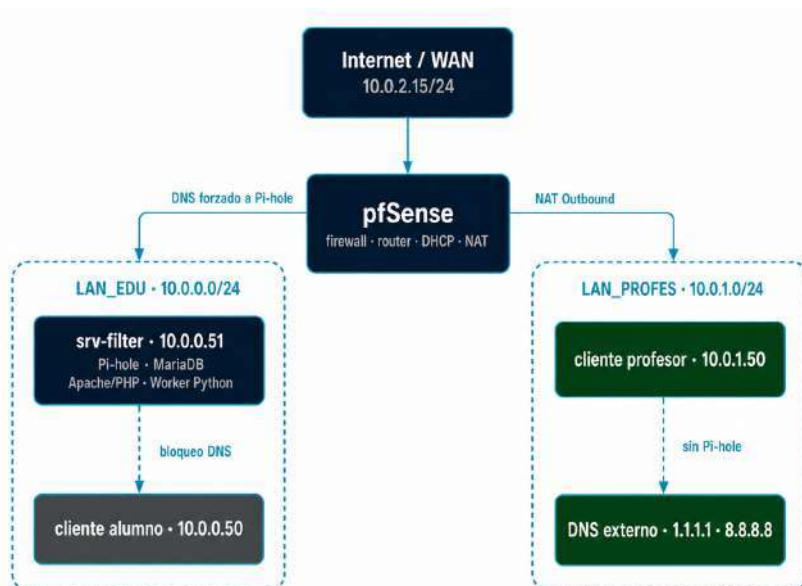


Figura 2. Arquitectura general d'EduNetGuard. Font: elaboració pròpia.

### 3.4 Base de dades MariaDB

La base de dades edunetguard és la font de veritat del sistema. Conté tota la lògica educativa del projecte: professors, assignatures, permisos, horaris, categories de bloqueig, dominis, regles actives i l'estat d'aplicació de cada bloqueig. Cap component del sistema modifica Pi-hole directament sense haver consultat primer MariaDB.

#### 3.4.1 Estructura de les taules principals

Taula	Camps principals	Funció
profesores	id, nombre, usuario, password (hash), es_admin, activo	Usuaris del sistema. Diferencia professors normals d'administradors.
asignaturas	id (varchar), nombre	Assignatures disponibles al centre. Identificadors com 0374, 0377, TUT.
profesor_asignatura	profesor_id, asignatura_id (PK composta)	Relaciona cada professor amb les assignatures que pot gestionar.
clases	id, nombre	Grups d'alumnes: ASIX_A, ASIX_B.
horarios	id, clase_id, asignatura_id, dia, hora_inicio, hora_fin	Franges horàries de cada assignatura per cada grup.
bloqueos	id, nombre	Categories de bloqueig predefinides (IA, JUEGOS, STREAMING, etc.).
dominios_bloqueo	id, bloqueo_id, dominio, tipo, añadido_por	Dominis associats a cada categoria de bloqueig.
reglas	id, bloqueo_id, horario_id, activa, activada_por	Relació entre una franja horària i una categoria de bloqueig activa.
regla_dominios_extra	id, horario_id, dominio, tipo, activada_por	Dominis addicionals bloquejats per un professor en una franja concreta.
permitidos_regla	id, horario_id, dominio, tipo, activada_por	Dominis permesos com a excepció, que prevalen sobre qualsevol bloqueig.
estado_bloqueos	id, bloqueo_id, aplicado, ultima_actualizacion	Registra si cada categoria EDUNET s'ha aplicat realment a Pi-hole.

Font: elaboració pròpia.

La taula reglas recull el que el professor ha configurat. La taula estado\_bloqueos recull el que el worker ha aplicat realment a Pi-hole. Aquesta separació és la base de la diferenciació visual de l'horari: una franja pot tenir configuració guardada sense estar activa en aquell moment.

## 3.5 Capa web i backend PHP

La interfície web consta de quatre pantalles principals en HTML, CSS i JavaScript. El backend PHP gestiona tota la lògica del servidor: validació de sessions, comprovació de permisos, accés a MariaDB i respostes en format JSON. La comunicació entre el frontend i el backend es fa via peticions HTTP amb dades JSON, cosa que separa clarament la part visual de la lògica del servidor.

### 3.5.1 Pantalles HTML

Fitxer	Funció
index.html	Pantalla inicial: login amb usuari i contrasenya, selector de grup (ASIX_A / ASIX_B), accés a horari personal i, si és admin, botó de panell administrador.
horario.html	Mostra l'horari del grup seleccionat o el personal del professor.
gestion-bloqueo.html	Configuració de bloquejos d'una franja concreta: activar categories, afegir webs extra bloquejades, definir excepcions permeses, guardar i eliminar configuració.
admin.html	Panell exclusiu per a l'administrador. Llista d'usuaris amb opcions d'edició, creació, canvi de contrasenya, activació/desactivació i eliminació. Taula de permisos per assignatura.

Font: elaboració pròpia.

El disseny detallat de cada pantalla, les decisions de disseny preses i el flux de navegació entre pantalles es descriuen a la secció 3.5.3.

### 3.5.2 Fitxers del backend PHP

Fitxer	Funció principal
config.php	Paràmetres de connexió a MariaDB (host, base, usuari, contrasenya). No s'ha de publicar ni mostrar amb credencials reals.
common.php	Funcions compartides: inicialització de sessió, connexió PDO, respostes JSON, validació de login, normalització de grups, dies, franges i dominis, comprovació de permisos.
login.php	Valida usuari i contrasenya, comprova que el compte estigui actiu i crea la sessió PHP amb el rol corresponent (professor o admin).
logout.php	Tanca la sessió de l'usuari i elimina la cookie de sessió.
me.php	Comprova la sessió activa, refresca les dades de l'usuari des de MariaDB i retorna rol, estat i permisos per assignatura.
get_ranges.php	Retorna les franges configurades i les realment actives per a un grup i dia. Permet mostrar la diferència entre «Configurada» i «Actiu» a l'horari.
get_config.php	Carrega la configuració guardada d'una franja: categories actives, dominis extra bloquejats i excepcions permeses.
save_config.php	Guarda una configuració de bloqueig. Normalitza dominis, comprova permisos, elimina la configuració anterior de la franja i desa categories, dominis extra i excepcions.
remove_config.php	Elimina tota la configuració associada a una franja: regles, dominis extra i excepcions permeses.
admin_data.php	Retorna al panell administrador la llista d'usuaris, assignatures disponibles i permisos actuals.
admin_user_save.php	Crea o modifica usuaris. Valida camps, manté proteccions sobre l'usuari administrador principal i desa contrasenyes noves amb password_hash (bcrypt).
admin_set_permissions.php	Actualitza els permisos d'un professor per assignatura, validant que les assignatures existeixin a la base de dades.
admin_user_delete.php	Desactiva o elimina usuaris. Gestiona dependències amb altres taules i detecta el nom real de la columna añadido_por via SHOW COLUMNS per evitar errors de codificació.

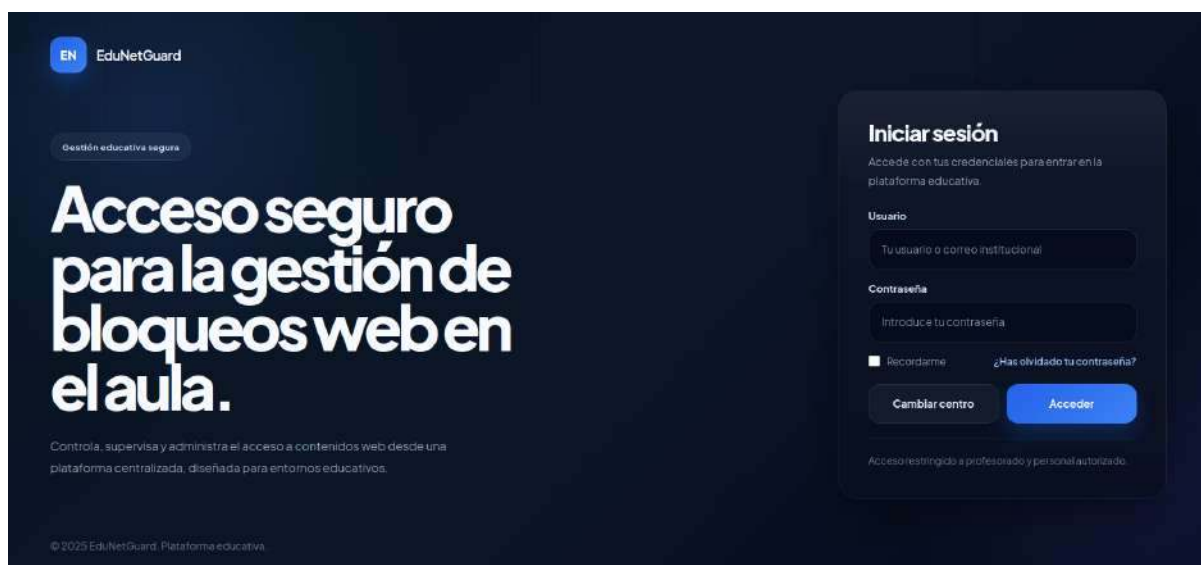
Font: elaboració pròpia. Rutes: /var/www/html/backend/

### 3.5.3 Disseny de la interfície i flux de navegació

La interfície web d'EduNetGuard està formada per quatre pantalles principals. El disseny s'ha orientat a la simplicitat d'ús: el professorat ha de poder configurar un bloqueig en el menor nombre de passos possible, sense necessitar coneixements tècnics de xarxes ni de sistemes.

#### Pantalla 1 — Login (index.html)

És la pantalla d'entrada al sistema. Permet iniciar sessió amb usuari i contrasenya. El backend valida que el compte existeixi i estigui actiu, i retorna el rol de l'usuari (professor o administrador). Segons el rol, la pantalla mostra el selector de grup per als professors o el botó d'accés al panell administrador per als usuaris amb permisos d'administració. També ofereix accés directe a l'horari personal del professor sense haver de seleccionar un grup.



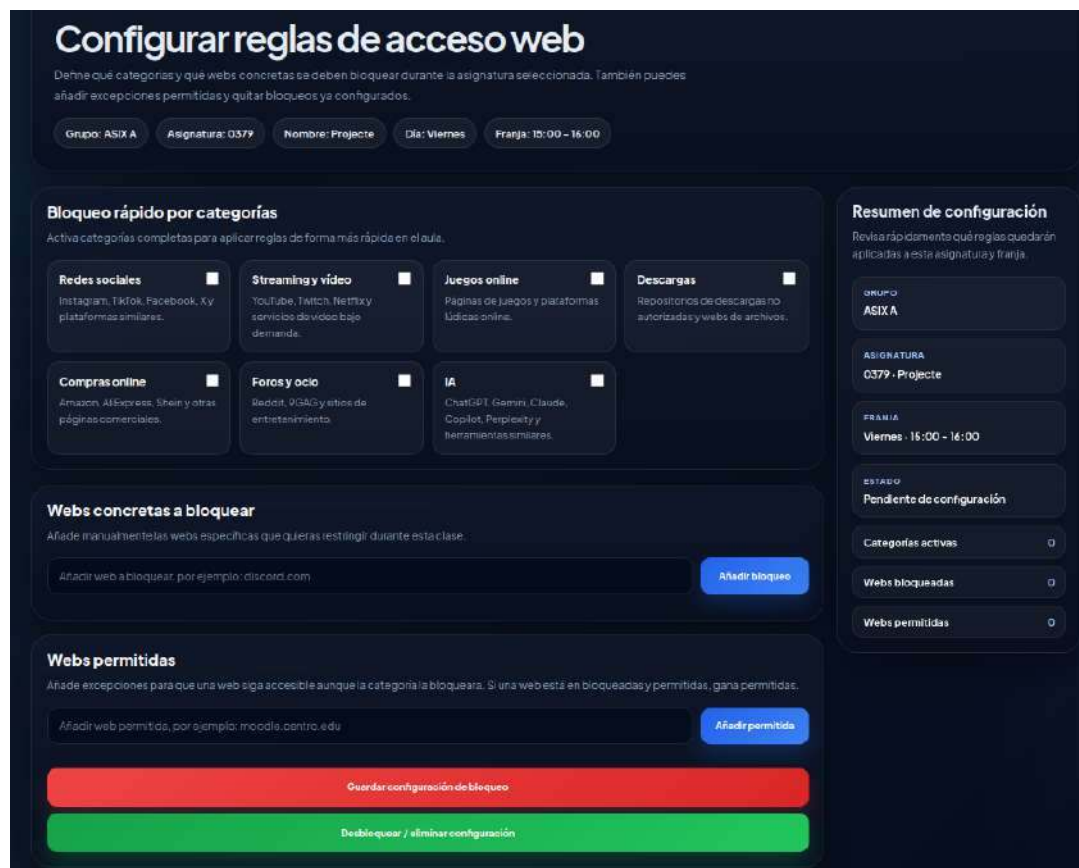
#### Pantalla 2 — Horari (horario.html)

Mostra l'horari del grup seleccionat o l'horari personal del professor. Cada cel·la de l'horari és una franja clicable. Una de les decisions de disseny més importants d'aquesta pantalla és la diferenciació visual entre franges: les franges amb configuració guardada es mostren en un color (gris), mentre que les franges actives en aquell moment es mostren en un altre (verd). Això permet al professor saber d'un cop d'ull què està passant sense necessitat d'entrar a cada franja. En fer clic sobre una franja, la navegació continua cap a la pantalla de gestió de bloquejos.



### Pantalla 3 — Gestió de bloquejos (gestion-bloqueo.html)

És la pantalla central per a la feina del professor. Permet activar categories de contingut (IA, Jocs, Xarxes socials, Streaming, etc.), afegir dominis concrets a bloquejar que no pertanyen a cap categoria, i declarar excepcions permeses que prevalen sobre qualsevol bloqueig actiu. Les tres seccions (categories, webs extra i webs permeses) estan clarament separades visualment per facilitar la comprensió. El botó de guardar envia la configuració al backend, que la desa a MariaDB; el worker l'aplicarà en menys d'un minut.



## Pantalla 4 — Panell administrador (admin.html)

Accessible només per a usuaris amb rol d'administrador. Presenta dues taules: la primera llista tots els professors amb el seu estat (actiu o inactiu) i ofereix les accions de crear, editar, canviar contrasenya, activar/desactivar i eliminar; la segona mostra i permet modificar els permisos per assignatura de cada professor. La decisió de mostrar les dues taules a la mateixa pantalla, en lloc de separar-les en pàgines diferents, permet a l'administrador veure i ajustar l'estat d'un professor i els seus permisos en una sola operació.

**Panel administrador**

### Gestión de profesorado y permisos

Crea profesores, modifica datos, cambia contraseñas, activa o desactiva usuarios y marca qué asignaturas puede gestionar cada profesor.

#### Usuarios

Selecciona un usuario para editarlo. Los administradores ven todo, los profesores dependen de sus permisos.

NOMBRE	USUARIO	ROL	ESTADO
SuperAdmin	admin	Admin	Activo
Daniel Martínez	dmartinez	Profesor	Activo
David García	dgarcia	Profesor	Activo
Iván Martínez	imartinez	Profesor	Activo
Jordi Mimó	jmimo	Profesor	Activo
Oscar Torrente	otorrente	Profesor	Activo

#### Nuevo usuario

Crea un profesor o selecciona uno de la tabla.

**Nombre completo**  
Ej: Profesor Sustituto

**Usuario**  
Ej: psustituto

**Contraseña nueva**  
Vacío = no cambiar

Usuario activo  
 Administrador

**Guardar usuario** **Nuevo**

**Desactivar** **Eliminar**

#### Permisos por asignatura

Selecciona un profesor para ver y cambiar sus permisos. **Guardar permisos**

No hay ningún usuario seleccionado.

## 3.6 Worker Python i automatització amb cron

El worker és l'script Python que fa que tot tingui sentit. La web guarda coses a MariaDB, però MariaDB per si sola no bloqueja res. Pi-hole bloqueja DNS, però no sap res d'horaris ni de professors. El worker llegeix MariaDB cada minut, mira si ara toca bloquejar alguna cosa, i actualitza Pi-hole en conseqüència. Sense el worker el sistema seria només una base de dades amb formularis.

### 3.6.1 Lògica d'execució del worker

Pas	Acció del worker
1	Connecta a MariaDB i llegeix totes les regles i configuracions guardades.
2	Obté el dia de la setmana i l'hora actual del sistema.
3	Calcula quines franges horàries coincideixen exactament amb el dia i l'hora actuals.
4	Per a cada categoria EDUNET, decideix si ha d'estar activa (enabled=1) o desactivada (enabled=0) a gravity.db.
5	Aplica la prioritat de les excepcions permeses: si un domini apareix a permesos_regla, s'afegeix a l'allowlist de Pi-hole independentment de les categories actives.
6	Aplica els dominis extra bloquejats de regla_dominios_extra com a entrades puntuals a gravity.db.
7	Executa pihole reloadlists per aplicar els canvis al filtratge DNS actiu.
8	Actualitza la taula estado_bloqueos a MariaDB per reflectir l'estat real aplicat.
9	Escriu el resultat al log /home/usuario/edunetguard-worker/sync_pihole_from_db.log.

Font: elaboració pròpia.

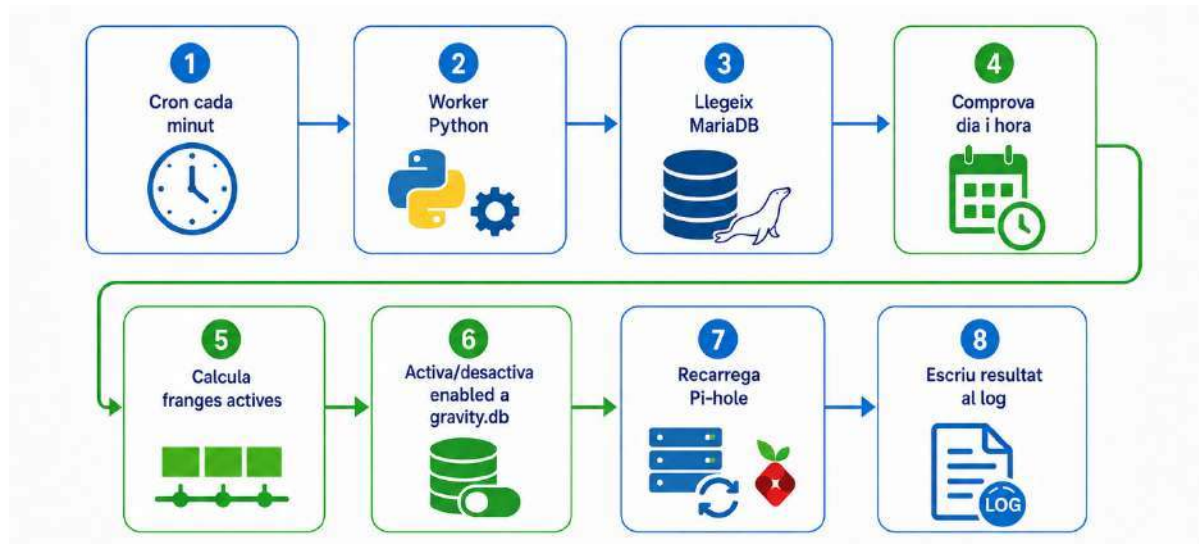


Figura 3. Funcionament del worker Python. Font : elaboració pròpia amb suport d'eines d'intel·ligència artificial per a la generació visual.

### 3.6.2 Automatització amb cron

El worker s'executa com a root cada minut via cron. Aquesta freqüència és suficient per al context del projecte: permet aplicar canvis poc després que el professor guardi la configuració i desactivar bloquejos quan acaba la franja sense que calgui cap intervenció manual.

La configuració exacta de la línia de cron es pot consultar a l'Annex C.

El log del worker és la principal eina de diagnòstic: permet comprovar si les regles s'han aplicat correctament, si hi ha errors de connexió a MariaDB o si Pi-hole no s'ha recarregat.

### 3.6.3 Script `seed_pihole_categories.py`

Abans de posar en marxa el worker per primera vegada, cal executar el script de seed. Aquest script crea totes les entrades EDUNET a `gravity.db` amb `enabled=0`, de manera que el worker pugui activar-les i desactivar-les sense crear-les de nou. El script fa una còpia de seguretat de `gravity.db` abans de modificar-lo i recarrega Pi-hole al final.

## 3.7 Panell administrador

El panell administrador és accessible únicament per a usuaris amb `es_admin=1`. Permet gestionar tots els aspectes dels comptes del sistema sense necessitat d'accedir directament a MariaDB.

Funcions disponibles:

- Veure la llista de tots els professors amb el seu estat (actiu/inactiu) i si són administradors.
- Crear un nou professor: nom complet, nom d'usuari, contrasenya (hashejada automàticament) i rol.
- Editar nom complet i nom d'usuari d'un professor existent.
- Canviar la contrasenya d'un professor (es desa sempre amb `password_hash`).
- Activar o desactivar un compte de professor sense eliminar-lo.
- Eliminar un professor, gestionant les dependències amb altres taules (horaris, regles, dominis).
- Assignar o revocar permisos per assignatura de forma individual.

El panell administrador no toca mai Pi-hole directament. Tota la gestió recau sobre la base de dades MariaDB, i el worker ja s'encarrega d'aplicar els canvis al filtratge DNS quan correspon.

### 3.8 Seguretat del sistema

La seguretat s'ha treballat en quatre nivells independents. Aquesta aproximació per capes és important perquè el sistema no depèn d'una única barrera: si una capa falla, les altres continuen actuant.

Nivell	Mesura aplicada	Detall
Xarxa	Separació LAN_EDU / LAN_PROFES	El professorat no pot ser afectat pels bloquejos aplicats als alumnes. Dues xarxes amb DNS i regles independents.
DNS	Redirect al port 53 cap a Pi-hole	pfSense força que tot el trànsit DNS de LAN_EDU vagi a Pi-hole, independentment del DNS configurat al client.
Aplicació	Sessió PHP i validació d'usuari actiu	Cada petició al backend comprova que hi ha una sessió vàlida i que l'usuari no ha estat desactivat.
Permisos	Permisos assignatura per	Cada professor només pot modificar franges de les assignatures que té assignades. El backend ho valida, no només la part visual.
Contrasenyes	password_hash / password_verify (bcrypt)	Les contrasenyes de tots els usuaris estan hashejades amb bcrypt. Quan es crea o modifica un usuari, el backend aplica sempre password_hash abans de desar-ho a la base de dades.
Accés admin	Rol es_admin=1	El panell administrador comprova explícitament el camp es_admin a MariaDB. No és suficient tenir sessió iniciada.

Font: elaboració pròpia.

Les limitacions de seguretat que encara queden pendents, com HTTPS, CSRF, DNS over HTTPS o còpies de seguretat automàtiques, es tracten a l'apartat 3.11 de limitacions i millores futures.

## 3.9 Problemes trobats i solucions aplicades

Durant el desenvolupament van aparèixer diversos problemes que van obligar a canviar decisions inicials. Aquests imprevistos són importants perquè expliquen per quin motiu el sistema final funciona d'aquesta manera i no d'una altra. També mostren el procés real de diagnòstic, proves i adaptació del projecte.

### 3.9.1 Disputa de ports entre Apache i Pi-hole

Un dels primers problemes va ser que Pi-hole ja utilitzava el port 80 per al seu panell web. Quan es va instal·lar Apache per servir EduNetGuard, els dos serveis intentaven utilitzar el mateix port, cosa que provocava conflicte.

La solució aplicada va ser mantenir Pi-hole al port 80 i configurar Apache perquè servís EduNetGuard al port 8080. D'aquesta manera no es modifica el funcionament habitual de Pi-hole i la web del projecte queda accessible a través de: <http://10.0.0.51:8080>

Aquesta decisió va quedar com a part del disseny final del sistema.

### 3.9.2 El client alumne no utilitzava Pi-hole com a DNS

Durant les primeres proves es va veure que el client no sempre utilitzava Pi-hole com a servidor DNS. Això feia que els bloquejos no s'apliquessin correctament, perquè les consultes DNS no passaven pel filtratge.

Per solucionar-ho, es va configurar el DHCP de pfSense perquè entregués 10.0.0.51 com a DNS a la xarxa LAN\_EDU. A més, es va afegir una redirecció DNS al port 53 cap a Pi-hole. Així, encara que un client intenti utilitzar un DNS extern, el trànsit DNS clàssic acaba passant per Pi-hole.

### 3.9.3 Desbloqueig poc fiable amb regles regex

Al principi es va provar crear i eliminar regles de Pi-hole cada vegada que un professor activava o desactivava un bloqueig. Aquest enfocament va donar problemes, sobretot amb entrades duplicades i desbloquejos que no sempre funcionaven bé.

La solució va ser canviar l'estratègia. En lloc de crear i eliminar regles contínuament, es van sembrar les categories EDUNET una vegada dins de gravity.db i el worker només canvia el camp enabled. Quan una categoria ha d'estar activa, queda amb enabled=1; quan no s'ha d'aplicar, queda amb enabled=0.

Aquest canvi va fer el sistema més estable i més fàcil de revisar.

### 3.9.4 Error pel mòdul php-mbstring

En una fase del desenvolupament va aparèixer un error relacionat amb la funció `mb_strtolower()`. El problema era que el mòdul `php-mbstring` no estava instal·lat a Ubuntu Server.

La solució va ser instal·lar aquest mòdul i reiniciar Apache. Després d'això, el backend PHP va poder treballar correctament amb funcions de text multibyte, necessàries per tractar cadenes amb accents o caràcters especials.

### 3.9.5 Confusió entre franges configurades i franges actives

Un altre problema va ser que la web mostrava una franja com si estigués bloquejada encara que en aquell moment no fos una franja activa. Això podia confondre el professorat, perquè no quedava clar si la franja només tenia una configuració guardada o si el bloqueig s'estava aplicant realment.

La solució va ser separar aquests dos conceptes al backend i a la interfície:

**Franja configurada:** té una regla guardada a MariaDB.

**Franja activa:** coincideix amb el dia i l'hora actuals i s'està aplicant.

A partir d'això, la web diferencia visualment les franges configurades de les actives.

### 3.9.6 El professorat no podia accedir a `srv-filter`

Quan es va crear la xarxa `LAN_PROFES`, el professorat tenia Internet i arribava correctament al gateway `10.0.1.1`, però no podia accedir al servidor `srv-filter` situat a `10.0.0.51`.

El problema estava en el retorn del trànsit. `srv-filter` no sabia tornar correctament cap a la xarxa `10.0.1.0/24`. Per no tocar rutes al servidor, es va decidir resoldre-ho des de `pfSense` amb una regla NAT Outbound.

La regla aplicada fa que el trànsit des de `LAN_PROFES` cap a `10.0.0.51` surti traduït amb l'adreça de la interfície LAN de `pfSense`. Això permet que el professorat accedeixi a la web sense quedar afectat pel filtratge DNS dels alumnes.

### 3.9.7 Error amb `anadido_por` i `añadido_por`

En el panell administrador va aparèixer un problema en eliminar usuaris per la diferència entre els noms de columna `anadido_por` i `añadido_por`. Aquesta diferència venia de la codificació i podia fer que el backend fallés si assumia un nom fix.

La solució va ser detectar el nom real de la columna amb `SHOW COLUMNS` abans de fer l'operació. No és la solució més elegant, però sí que és pràctica i robusta per a l'estat real de la base de dades.

### 3.9.8 Reset States inestable per SSH

Es va estudiar la possibilitat d'utilitzar Reset States de pfSense per tallar connexions actives quan s'aplicava un bloqueig. La idea era evitar que una connexió ja oberta continués funcionant després d'activar una restricció.

Es van fer proves amb SSH i pfctl, però en alguns casos el procés es quedava penjat o donava timeout. Per això es va decidir no fer dependre el funcionament principal del projecte d'aquesta funció.

Finalment, Reset States queda documentat com una millora parcial i opcional. El sistema funciona correctament per a noves connexions sense necessitar aquesta part.

### 3.9.9 Errors inicials d'instal·lació de pfSense

Durant la instal·lació de pfSense van aparèixer problemes relacionats amb la configuració de VirtualBox, com arquitectura de la màquina virtual, mode de 64 bits, boot order, ZFS i controladora de disc.

La solució va ser ajustar la configuració de la màquina virtual i utilitzar pfSense CE. Aquest procés va servir per deixar documentats els requisits mínims de la VM abans de muntar la resta de serveis.

### 3.9.10 Eliminació d'usuaris amb historial

En el panell administrador, eliminar un usuari directament podia fallar si aquell professor tenia dades relacionades en altres taules, com regles o dominis afegits. El problema venia de les restriccions de clau forana de MariaDB.

La solució va ser modificar `admin_user_delete.php` perquè gestionés aquestes dependències abans d'eliminar o desactivar l'usuari. Això va fer el backend de l'administrador més segur i coherent amb el model de dades.

### 3.10 Proves i validació

No ens interessava que `systemctl status apache2` digués "active". Ens interessava que quan un professor guardés un bloqueig des de la web, l'alumne deixés d'accedir al domini en menys d'un minut. Totes les proves estan orientades a demostrar exactament això, d'extrem a extrem.

El recorregut complet validat és: el professor inicia sessió, configura un bloqueig, MariaDB desa la regla, el worker aplica el canvi, Pi-hole activa el bloqueig, l'alumne queda filtrat i el professor continua navegant sense filtratge des de LAN\_PROFES.

Prova	Objectiu	Procediment	Resultat
Interfícies pfSense	Validar WAN, LAN_EDU i LAN_PROFES operatives	Revisar configuració d'interfícies al WebConfigurator	Tres xarxes diferenciades i actives. Correcte.
DHCP LAN_EDU	Client alumne rep Pi-hole com a DNS	Connectar client a LAN_EDU i revisar configuració rebuda amb <code>ip route</code> i <code>resolvectl</code>	DNS 10.0.0.51 entregat. Correcte.
DNS redirect	Port 53 forçat cap a Pi-hole	Des del client, dig <code>@1.1.1.1</code> retorna resposta de Pi-hole	Redirecció activa. Correcte.
NAT Outbound PROFES	Professor arriba a la web des de LAN_PROFES	<code>ping 10.0.0.51</code> i <code>curl -I http://10.0.0.51:8080</code> des de 10.0.1.50	Accés a la web correcte. Correcte.
Login web	Autenticació i sessió PHP funcionen	Login amb usuari actiu, verificació de rol i accés al panell	Sessió creada correctament per a professor i admin. Correcte.
Horari personal	Professor veu només les seves assignatures	Entrar amb jmimo i comprovar que apareixen únicament 0377 i 0379	Horari filtrat per permisos. Correcte.
Gestió de bloquejos	Categories, extra i permeses es guarden	Guardar configuració amb categoria IA i un domini extra des de la web	Configuració present a MariaDB. Correcte.
Worker Python	El worker aplica canvis automàticament	Revisar log del worker 60 s després de guardar una regla	Canvi detectat i aplicat a Pi-hole. Correcte.
gravity.db	<code>enabled=1/0</code> canvia correctament	<code>sqlite3 gravity.db</code> i consultar entrades EDUNET	Camp <code>enabled</code> coherent amb la franja activa. Correcte.
Client alumne filtrat	Alumne no pot accedir a domini bloquejat	Provar accés a <code>instagram.com</code> des de LAN_EDU amb bloqueig actiu	Domini bloquejat per Pi-hole. Correcte.
Professor sense filtratge	LAN_PROFES no queda afectada pels bloquejos	Provar accés als mateixos dominis des de LAN_PROFES durant el bloqueig	Navegació lliure per al professor. Correcte.

Prova	Objectiu	Procediment	Resultat
Cron actiu	Cron executa el worker cada minut	grep CRON /var/log/syslog i revisar timestamps del log	Execució cada minut confirmada. Correcte.
Panell administrador	Admin pot crear, editar i eliminar usuaris	Crear usuari nou, assignar permisos, desactivar i eliminar	Totes les operacions funcionen. Correcte.
Reset States pfSense	Tallar connexions actives en canviar bloqueig	Proves SSH/pfctl des de srv-filter a pfSense	Resultat parcial: alguns intents es penguen. Parcial.

Font: elaboració pròpia. Les captures de pantalla de les proves s'inclouen als Annexos com a material de suport per a la defensa.

### Configuració de bloqueig des de la web, mostrant una franja i les categories seleccionades.

The screenshot displays the EduNetGuard web interface for configuring network filtering. It features several sections:

- Activar categorías completas para aplicar reglas de forma más rápida en el aula:** A grid of category cards with checkboxes:
  - Redes sociales** (checked): Instagram, TikTok, Facebook, X y plataformas similares.
  - Streaming y video** (unchecked): YouTube, Twitch, Netflix y servicios de video bajo demanda.
  - Juegos online** (unchecked): Páginas de juegos y plataformas lúdicas online.
  - Descargas** (unchecked): Repositorios de descargas no autorizadas y webs de archivos.
  - Compras online** (unchecked): Amazon, AliExpress, Shein y otras páginas comerciales.
  - Foros y ocio** (unchecked): Reddit, 9GAG y sitios de entretenimiento.
  - IA** (unchecked): ChatGPT, Gemini, Claude, Copilot, Perplexity y herramientas similares.
- Resumen de configuración:** A summary panel on the right showing:
  - GRUPO: ASIX A
  - ASIGNATURA: 0379 - Proyecto
  - FRANJA: Viernes - 15:00 - 16:00
  - ESTADO: Configuración guardada
  - Categorías activas: 1
  - Webs bloqueadas: 1
  - Webs permitidas: 0
- Webs concretas a bloquear:** A section for manually adding specific websites to be blocked, with an input field containing "forocoche.com" and an "Añadir bloqueo" button.
- Webs permitidas:** A section for manually adding specific websites to be allowed, with an input field containing "moodle.centro.edu" and an "Añadir permitida" button.
- Buttons:** "Guardar configuración de bloqueo" (red), "Desbloquear / eliminar configuración" (green), and "Configuración guardada para 0379 - Proyecto (Viernes, 15:00 - 16:00)." (green).

Regla desada a MariaDB després de desar la configuració.

```

Last login: Fri May 15 15:10:38 2026
usuario@filter:~$ mariadb -u edunetguard -p edunetguard -e "
SELECT r.id, r.bloqueo_id, b.nombre AS bloqueo, r.horario_id, r.activa, r.activada_por
FROM reglas r
JOIN bloqueos b ON r.bloqueo_id = b.id
ORDER BY r.id DESC
LIMIT 10;
"
Enter password:
+-----+-----+-----+-----+-----+-----+
| id | bloqueo_id | bloqueo          | horario_id | activa | activada_por |
+-----+-----+-----+-----+-----+-----+
| 222 | 3 | REDES_SOCIAL    | 58 | 1 | 2 |
| 221 | 3 | REDES_SOCIAL    | 18 | 1 | 2 |
| 206 | 3 | REDES_SOCIAL    | 47 | 1 | NULL |
| 205 | 3 | REDES_SOCIAL    | 7 | 1 | NULL |
| 204 | 3 | REDES_SOCIAL    | 46 | 1 | NULL |
| 203 | 3 | REDES_SOCIAL    | 6 | 1 | NULL |
| 202 | 1 | IA              | 41 | 1 | 1 |
| 201 | 1 | IA              | 1 | 1 | 1 |
| 200 | 7 | COMPRAS_ONLINE | 41 | 1 | 1 |
| 199 | 7 | COMPRAS_ONLINE | 1 | 1 | 1 |
+-----+-----+-----+-----+-----+-----+

```

```

usuario@filter:~$ mariadb -u edunetguard -p edunetguard -e "
SELECT * FROM regla_dominios_extra ORDER BY id DESC LIMIT 10;
"
Enter password:
+-----+-----+-----+-----+-----+-----+
| id | horario_id | dominio          | tipo | activada_por | created_at |
+-----+-----+-----+-----+-----+-----+
| 74 | 58 | forocoches.com | regex | 2 | 2026-05-15 15:31:10 |
| 73 | 18 | forocoches.com | regex | 2 | 2026-05-15 15:31:10 |
+-----+-----+-----+-----+-----+-----+

```

Log del worker aplicant els canvis sobre Pi-hole.

```

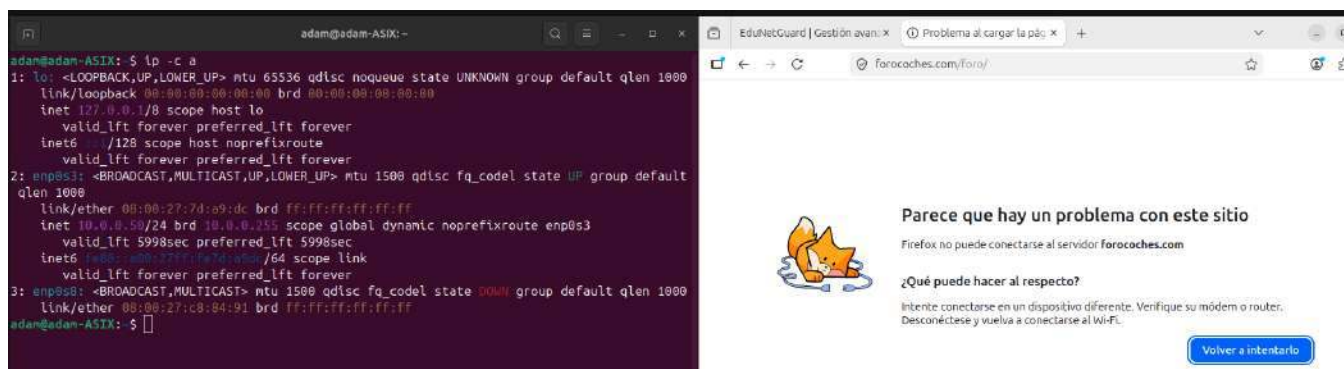
[INFO] Día 5 Hora 15:27:01
[=] Sin cambios
[INFO] Día 5 Hora 15:28:01
[=] Sin cambios
[INFO] Día 5 Hora 15:29:01
[=] Sin cambios
[INFO] Día 5 Hora 15:30:01
[=] Sin cambios
[INFO] Día 5 Hora 15:31:01
[=] Sin cambios
[INFO] Día 5 Hora 15:32:01
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)tiktok\.com$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)instagram\.com$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)cdninstagram\.com$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)x\.com$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)twitter\.com$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)twimg\.com$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)facebook\.com$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)fbcdn\.net$ -> 1
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)pinterest\.com$ -> 1
[CAMBIO] EDUNET:FOROS_OCIO | (\.|^)forocoches\.com$ -> 1
[OK] Pi-hole actualizado
[INFO] Día 5 Hora 15:33:01
[=] Sin cambios
usuario@filter:~$

```

Consulta de gravity.db mostrant enabled=1/0 a les entrades EDUNET.

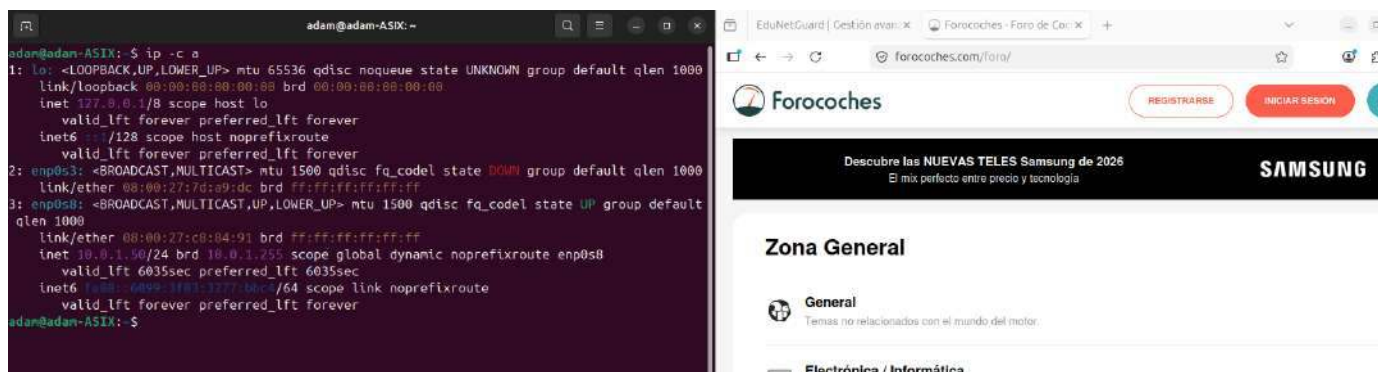
```
usuari@filter:~$ sudo sqlite3 /etc/pihole/gravity.db "
SELECT comment, type, MIN(enabled), MAX(enabled), COUNT(*)
FROM domainlist
WHERE comment LIKE 'EDUNET:%'
GROUP BY comment, type
ORDER BY comment;
"
EDUNET:ALLOW|2|0|0|2
EDUNET:COMPRAS_ONLINE|3|0|0|7
EDUNET:DESCARGAS|3|0|0|6
EDUNET:EXTRA_BLOCK|3|0|0|3
EDUNET:FOROS_OCIO|3|0|1|4
EDUNET:IA|3|0|0|15
EDUNET:JUEGOS|3|0|0|9
EDUNET:REDES_SOCIAL|3|1|1|9
EDUNET:STREAMING|3|0|0|14
```

Prova del client alumne amb el domini bloquejat.



The image shows a terminal window on the left and a browser window on the right. The terminal window displays the output of the command `ip -c a`, showing network configuration for interfaces `lo`, `enp0s3`, and `enp0s8`. The browser window shows a Firefox error message: "Parece que hay un problema con este sitio" (It seems there is a problem with this site). The message states: "Firefox no puede conectarse al servidor forocoches.com" (Firefox cannot connect to the server forocoches.com). Below the message, there is a "Volver a intentarlo" (Try again) button.

Prova del client professor des de LAN\_PROFES sense filtratge DNS.



The image shows a terminal window on the left and a browser window on the right. The terminal window displays the output of the command `ip -c a`, showing network configuration for interfaces `lo`, `enp0s3`, and `enp0s8`. The browser window shows the website `forocoches.com` successfully loaded. The website header includes the text "Descubre las NUEVAS TELES Samsung de 2026" (Discover the NEW Samsung TELES of 2026) and "El mix perfecto entre precio y tecnología" (The perfect mix between price and technology). Below the header, there is a "Zona General" section with a "General" category and a "Electrónica / Informática" category.

### 3.11 Limitacions i millores futures

La solució funciona correctament en un entorn de laboratori, però no s'ha de presentar com si fos un sistema tancat per a producció. Durant el desenvolupament s'han detectat algunes limitacions que cal tenir en compte.

La primera és que el filtratge DNS controla les consultes DNS clàssiques, però no cobreix completament casos com DNS over HTTPS. Aquest mecanisme permet que algunes aplicacions facin consultes DNS xifrades per HTTPS i no utilitzin el DNS local del centre. Aquesta limitació no invalida el projecte, però en un entorn real s'hauria de controlar amb polítiques de navegador o regles de tallafoc.

Una altra limitació és que `srv-filter` concentra diversos serveis importants: Pi-hole, MariaDB, Apache, PHP i el worker Python. Per al laboratori és una solució senzilla i suficient, perquè facilita la instal·lació i les proves. Tot i això, en producció seria més recomanable separar alguns serveis o, com a mínim, preparar una política de còpies de seguretat completa.

També queda pendent confirmar i aplicar mesures de seguretat addicionals, com HTTPS, protecció CSRF i còpies automàtiques. HTTPS permetria xifrar les credencials i les sessions web, CSRF ajudaria a protegir els formularis del backend i les còpies automàtiques reduirien el risc de perdre configuració o dades importants.

Aquestes limitacions no impedeixen que el sistema compleixi els objectius plantejats en el laboratori. El que indiquen és que, abans d'una implantació real en un centre educatiu, caldria reforçar la seguretat, el manteniment i la disponibilitat del sistema.

### 3.12 Relació transversal amb el cicle

EduNetGuard és un projecte transversal que integra coneixements de diverses assignatures del cicle d'ASIX. Aquesta transversalitat no és casual: la naturalesa del sistema obliga a combinar competències d'àmbits molt diferents per fer funcionar un conjunt de components que depenen els uns dels altres.

En l'àmbit de les xarxes, el projecte aplica adreçament IP, subxarxes, gateway, NAT, DHCP, DNS, redirecció de ports i segmentació de xarxes amb `pfSense`. La creació de `LAN_EDU` i `LAN_PROFES` com a xarxes separades, amb comportament de DNS diferent, és una aplicació directa dels continguts de Serveis de Xarxa i Internet (0375) i Administració de Sistemes Operatius (0374).

En l'àmbit dels sistemes, s'ha desplegat i administrat Ubuntu Server com a màquina de serveis, s'han instal·lat i configurat Apache, MariaDB i Pi-hole, i s'ha automatitzat l'execució del worker amb cron. El diagnòstic de problemes amb `systemctl`, `journalctl` i logs de sistema correspon als continguts d'Administració de Sistemes Operatius (0374).

En l'àmbit de les bases de dades, s'ha dissenyat un model relacional complet amb 11 taules, claus primàries, claus forànies, restriccions d'integritat i consultes SQL amb JOIN, INSERT ... ON DUPLICATE KEY UPDATE i subconsultes. Correspon directament als continguts de Gestió de Bases de Dades (0377).

En l'àmbit del desenvolupament web, la interfície combina HTML, CSS i JavaScript al frontend amb PHP al backend, utilitzant sessions, peticions HTTP amb JSON i validació de permisos. El worker Python gestiona connexions a dues bases de dades simultànies (MariaDB via PyMySQL i Pi-hole via `sqlite3`) amb tractament d'errors i logs. Correspon als continguts d'Implantació d'Aplicacions Web (0376).

En l'àmbit de la seguretat, s'han aplicat hash `bcrypt` per a contrasenyes, separació de xarxes, validació de sessions i rols, control de permisos per assignatura i una anàlisi

honestament de les limitacions (HTTPS, CSRF, DoH). Correspon als continguts de Seguretat i Alta Disponibilitat (0378).

Finalment, el projecte en si mateix és el mòdul de Projecte (0379): integra tots els àmbits anteriors en un sistema real, defensable i documentat, que resol un problema concret d'un centre educatiu amb programari lliure i decisions tècniques justificades.

## 4. CONCLUSIONS

### 4.1 Conclusions generals

EduNetGuard demostra que és possible construir una plataforma de control web educatiu funcional utilitzant exclusivament programari lliure i una arquitectura per capes. El projecte va més enllà d'instal·lar Pi-hole: afegeix una capa de gestió completa perquè el professorat pugui aplicar bloquejos de forma contextual, sense necessitar coneixements tècnics de xarxes ni d'administració de sistemes.

El que més ens va costar entendre al principi és que cap de les eines fa res sola. pfSense no sap res de professors ni d'horaris. Pi-hole no sap qui ha iniciat sessió. MariaDB no bloqueja res per si sola. Connectar-ho tot i fer que funcionés de manera coordinada és el que realment va portar temps i va generar la majoria dels problemes documentats a la secció 3.9.

La decisió de separar la xarxa del professorat ha estat un dels encerts més clars del projecte. Sense LAN\_PROFES, el professor quedaria afectat pels mateixos bloquejos que aplica als alumnes, cosa que faria inviable l'ús pràctic del sistema. Aquesta millora va aparèixer com a resposta a un problema real, no estava planificada des del principi.

Quan vam acabar de muntar-ho tot i vam veure que el professor podia bloquejar Instagram des de la web i l'alumne deixava d'accedir-hi en menys d'un minut, vam entendre que el projecte havia funcionat. No perquè fos perfecte, sinó perquè resolía exactament el que ens havíem proposat.

## 4.2 Consecució dels objectius

Objectiu	Estat	Comentari
Laboratori virtual amb pfSense, servidor i client	Assolit	Les tres màquines funcionen correctament sobre VirtualBox amb les dues xarxes diferenciades.
Filtratge DNS per a alumnes amb Pi-hole	Assolit	LAN_EDU usa Pi-hole i el DNS redirect de pfSense garanteix que tots els clients el facin servir.
Xarxa separada per al professorat	Assolit	LAN_PROFES usa DNS extern i no queda filtrada per Pi-hole. L'accés a la web es resol amb NAT Outbound.
Web de gestió per al professorat	Assolit	Permet login, consulta d'horaris i configuració de bloquejos amb persistència a MariaDB.
Panell administrador	Assolit	Gestiona usuaris, contrasenyes, estat de comptes i permisos per assignatura.
Worker Python i cron	Assolit	Sincronitza MariaDB amb Pi-hole cada minut de forma automàtica.
Dominis extra i excepcions permeses	Assolit	La lògica és correcta i s'integra al worker.
Diferenciació visual configurada/actiu	Assolit	La web mostra clarament l'estat de cada franja sense ambigüitat.
Reset States de pfSense	Parcial	S'ha analitzat i provat, però no ha quedat prou estable per dependre-hi.
HTTPS, CSRF, backups, mitigació DoH	Pendent	Queden com a millores futures prioritàries per a un entorn de producció.

Font: elaboració pròpia.

Els elements parcials o pendents no invaliden el projecte al laboratori. S'han documentat perquè serien imprescindibles en un desplegament real.

### 4.3 Valoració de la metodologia i planificació

La metodologia incremental ha estat adequada per a un projecte amb tants components relacionats entre si. Treballar per fases ha permès detectar errors de disseny aviat, quan el cost de corregir-los era baix, i adaptar les decisions a mesura que apareixien nous problemes reals.

La planificació inicial va evolucionar bastant durant el procés, i creiem que això és el més honest que podem dir. LAN\_PROFES, per exemple, no estava planificada des del principi: va aparèixer quan vam comprovar durant les proves que el professor quedava bloquejat pels seus propis filtres i no podia ni accedir a la web del projecte. El panell administrador va néixer d'una necessitat similar: calia una forma de gestionar usuaris sense tocar directament la base de dades cada vegada. La decisió d'usar `enabled=1/0` a Pi-hole en lloc de crear i eliminar regles contínuament va sorgir després d'hores depurant per quin motiu els desbloquejos no funcionaven de forma fiable. Cap d'aquestes millores estava al disseny original, però totes van aparèixer per un motiu real i documentat.

Si haguéssim de dir honestament quina assignatura hem aplicat més, seria difícil triar-ne una. Hem configurat pfSense com a Serveis de Xarxa, hem administrat Ubuntu Server com a Administració de Sistemes, hem dissenyat 11 taules amb claus forànies com a Gestió de Bases de Dades, i hem depurat sessions PHP i scripts Python com a Implantació d'Aplicacions Web. No ha estat un projecte d'una assignatura. Ha estat un projecte de tot el cicle alhora.

### 4.4 Visió de futur

Si continuéssim el projecte, les prioritats estarien clares: primer seguretat, després manteniment, i finalment ampliar funcionalitats.

En l'àmbit de la seguretat, les millores més importants serien implementar HTTPS, afegir protecció CSRF als formularis i millorar la mitigació de DNS over HTTPS. Aquestes mesures serien especialment importants si el sistema es volgués portar a un entorn real fora del laboratori.

En l'àmbit del manteniment, seria necessari completar una política de còpies de seguretat automàtiques. Aquestes còpies haurien d'incloure MariaDB, `gravity.db`, els fitxers web i els scripts del worker. També seria útil afegir logs més detallats, auditoria d'accions administratives i alguna forma de monitorització dels serveis principals.

Com a ampliació funcional, una millora interessant seria afegir un sistema d'importació d'horaris mitjançant fitxers CSV. Això permetria carregar l'horari d'un grup o cicle complet i crear o actualitzar assignatures, professors relacionats i permisos de forma més ràpida. Aquesta funció reduiria molt la feina manual inicial si el projecte s'hagués d'adaptar a altres grups o centres.

A nivell d'infraestructura, en un entorn de producció les xarxes de VirtualBox es podrien substituir per VLANs reals gestionades pel centre. També es podria valorar separar alguns serveis de `srv-filter` en màquines o contenidors diferents per millorar la disponibilitat.

Finalment, Reset States de pfSense es manté com una possible millora opcional. Es va analitzar durant el projecte, però no va quedar prou estable per formar part del funcionament principal. En una versió futura es podria recuperar si es troba una forma fiable de tallar connexions actives sense afectar la resta del sistema.

## 5. GLOSSARI

Terme	Definició
Backend	Part del sistema que processa les peticions de la web, valida sessions i guarda o consulta dades a MariaDB. En EduNetGuard correspon als fitxers PHP del directori /backend.
Categoria de bloqueig	Grup de dominis relacionats (IA, Jocs, Xarxes socials, etc.) que es pot activar en bloc per a una franja horària.
Cron	Servei de planificació de Linux que executa el worker Python cada minut de forma automàtica, sense intervenció manual.
DNS over HTTPS / DoH	Mecanisme que xifra les consultes DNS dins de connexions HTTPS, la qual cosa pot evitar el DNS local si no es controla a nivell de tallafoc o navegador.
DNS redirect	Regla de pfSense que redirigeix tot el trànsit DNS del port 53 de LAN_EDU cap a Pi-hole, independentment del DNS que el client tingui configurat.
enabled (camp a gravity.db)	Camp de la taula domainlist de Pi-hole que activa (1) o desactiva (0) una entrada de filtratge sense necessitat d'eliminar-la.
Excepció permesa	Domini declarat explícitament com a permès per a una franja, que preval sobre qualsevol categoria bloquejada activa.
Font de veritat	Concepte que indica que MariaDB conté la configuració real del sistema. Pi-hole aplica el filtratge, però el worker sempre llegeix MariaDB per decidir l'estat correcte.
Franja activa	Franja horària que coincideix amb el dia de la setmana i l'hora actuals, i per tant té bloquejos aplicats en aquell moment.
Franja configurada	Franja que té una configuració guardada a MariaDB, però que pot no ser activa en aquell moment perquè no coincideix amb el dia o l'hora.
gravity.db	Base de dades SQLite interna de Pi-hole on es guarden tots els dominis i les regles de filtratge. En EduNetGuard el worker la modifica directament.
Horari personal	Vista filtrada de l'horari que mostra únicament les franges de les assignatures assignades al professor que ha iniciat sessió.
LAN_EDU	Xarxa d'alumnes (10.0.0.0/24) amb DNS filtrat per Pi-hole i redirecció forçada al port 53.
LAN_PROFES	Xarxa del professorat (10.0.1.0/24) amb DNS extern i sense filtratge DNS dels alumnes.
NAT Outbound	Regla de pfSense que permet a LAN_PROFES accedir a srv-filter (10.0.0.51) emmascarant l'adreça d'origen, sense necessitat de modificar les rutes del servidor.
Panell administrador	Secció de la web accessible únicament per a administradors (es_admin=1). Permet gestionar usuaris, comptes i permisos per assignatura.
password_hash / password_verify	Funcions de PHP que generen i validen hashcs bcrypt per a les contrasenyes dels usuaris, evitant emmagatzemar-les en text pla.

Terme	Definició
pfSense	Distribució de FreeBSD per a tallafoc i router. En EduNetGuard actua com a gateway, DHCP, NAT, DNS redirect i separació de xarxes.
Pi-hole	Servidor DNS filtrant que aplica els bloquejos als alumnes. En EduNetGuard no el gestiona el professorat directament, sinó el worker.
Regla EDUNET	Entrada preparada a gravity.db per a una categoria de bloqueig d'EduNetGuard, identificada per un comentari que comença per EDUNET:.
Reset States	Funció de pfSense per eliminar estats de connexió actius. En EduNetGuard queda com a funcionalitat parcial i opcional.
SGBD	Sistema gestor de bases de dades. En EduNetGuard s'utilitza MariaDB per a la base principal i SQLite internament a Pi-hole.
srv-filter	Servidor Ubuntu principal que allotja Pi-hole, MariaDB, Apache, PHP, el worker Python i cron. IP: 10.0.0.51.
Sessió PHP	Mecanisme que manté l'estat d'autenticació entre peticions HTTP. El backend valida la sessió a cada operació per garantir que l'usuari és qui diu ser.
Worker Python	Script sync_pihole_from_db.py que llegeix les regles de MariaDB, calcula les franges actives i actualitza gravity.db de Pi-hole. S'executa cada minut via cron.

Font: elaboració pròpia.

## 6. BIBLIOGRAFIA

- [1] Netgate. pfSense Documentation. Disponible a: <https://docs.netgate.com/pfsense/en/latest/>
- [2] Pi-hole. Pi-hole Documentation. Disponible a: <https://docs.pi-hole.net/>
- [3] MariaDB Foundation. MariaDB Server Documentation. Disponible a: <https://mariadb.com/docs/server>
- [4] The Apache Software Foundation. Apache HTTP Server Documentation 2.4. Disponible a: <https://httpd.apache.org/docs/2.4/>
- [5] PHP Documentation Group. PHP Manual: password\_hash. Disponible a: <https://www.php.net/manual/en/function.password-hash.php>
- [6] PHP Documentation Group. PHP Manual: password\_verify. Disponible a: <https://www.php.net/manual/en/function.password-verify.php>
- [7] Python Software Foundation. Python 3 Documentation: sqlite3 module. Disponible a: <https://docs.python.org/3/library/sqlite3.html>
- [8] PyMySQL. PyMySQL Documentation. Disponible a: <https://pymysql.readthedocs.io/>
- [9] Ubuntu. Ubuntu Server Documentation. Disponible a: <https://ubuntu.com/server/docs>
- [10] Oracle. Oracle VirtualBox Documentation. Disponible a: <https://docs.oracle.com/en/virtualization/virtualbox/>
- [11] Agència de Ciberseguretat de Catalunya. (2023). Guia de seguretat per a centres educatius. Disponible a: <https://ciberseguretat.gencat.cat>
- [12] INCIBE. (2023). Ciberseguridad en el sector educativo. Institut Nacional de Ciberseguretat d'Espanya. Disponible a: <https://www.incibe.es/empresas/blog/ciberseguridad-en-el-sector-educativo>
- [13] ENISA. (2022). Cybersecurity Education Initiatives in the EU Member States. European Union Agency for Cybersecurity. Disponible a: <https://www.enisa.europa.eu/publications/cybersecurity-education-initiatives-in-the-eu-member-states>
- [14] ENISA. (2024). Cybersecurity Education Maturity Assessment. European Union Agency for Cybersecurity. Disponible a: <https://www.enisa.europa.eu/publications/cybersecurity-education-maturity-assessment>

## 7. ANNEXOS

Els annexos recullen informació de suport per a la defensa i el manteniment del projecte. S'han separat del cos principal perquè no pertanyen al flux de la memòria, sinó que serveixen com a referència tècnica ràpida.

### Annex A. Màquines virtuals del laboratori

Màquina	Sistema	IP / Xarxa	RAM / CPU / Disc	Funció
fw-edunetids	pfSense CE 2.8.1	WAN: 10.0.2.15 LAN: 10.0.0.1 PROFES: 10.0.1.1	1 GB / 1 nucli / 20 GB	Gateway, tallafoc, DHCP, NAT i DNS redirect
srv-filter	Ubuntu Server 22.04	LAN: 10.0.0.51 Pont: 192.168.254.83	2 GB / 2 nuclis / 30 GB	Pi-hole, MariaDB, Apache, PHP i worker Python
Client de proves	Ubuntu Desktop 22.04	Alumne: 10.0.0.50 Professor: 10.0.1.50	2 GB / 2 nuclis / 25 GB	Client per simular alumne i professor

Font: elaboració pròpia.

### Annex B. Xarxes, adreces IP i ports

Xarxa	Rang	Gateway	DNS	Rol
LAN_EDU	10.0.0.0/24	10.0.0.1	10.0.0.51 (Pi-hole)	Alumnes filtrats
LAN_PROFES	10.0.1.0/24	10.0.1.1	1.1.1.1 / 8.8.8.8	Professorat sense filtratge
WAN pfSense	10.0.2.15/24	VirtualBox NAT	Entorn	Sortida a Internet

Font: elaboració pròpia.

Port	Servei	Ús al projecte
53 TCP/UDP	DNS Pi-hole	Resolució DNS filtrada per als alumnes
80 TCP	Pi-hole web	Panell web intern de Pi-hole
8080 TCP	EduNetGuard (Apache)	Interfície web del projecte
443 TCP	pfSense WebConfigurator	Administració de pfSense
22 TCP	SSH	Administració remota (si està actiu)

Font: elaboració pròpia.

## Annex C. Rutes principals del projecte

Web (HTML):

- /var/www/html/index.html
- /var/www/html/horario.html
- /var/www/html/gestion-bloqueo.html
- /var/www/html/admin.html

Backend (PHP):

- /var/www/html/backend/config.php
- /var/www/html/backend/common.php
- /var/www/html/backend/login.php | logout.php | me.php
- /var/www/html/backend/get\_config.php | save\_config.php | remove\_config.php | get\_ranges.php
- /var/www/html/backend/admin\_data.php | admin\_user\_save.php | admin\_set\_permissions.php | admin\_user\_delete.php

Worker i scripts Python:

- /home/usuario/edunetguard-worker/sync\_pihole\_from\_db.py
- /home/usuario/edunetguard-worker/seed\_pihole\_categories.py

Logs:

- /home/usuario/edunetguard-worker/sync\_pihole\_from\_db.log (worker)
- /var/log/apache2/error.log (Apache/PHP)
- /var/log/syslog (cron i sistema)

Pi-hole i Apache:

- /etc/pihole/gravity.db | /etc/pihole/gravity.db.bak.\*
- /etc/apache2/ports.conf | /etc/apache2/sites-available/000-default.conf

## Annex D. Evidències de validació del sistema

Verificar que el worker s'executa amb cron:

```
sudo crontab -l  
grep CRON /var/log/syslog | tail -10
```

```
usuario@filter:~$ grep CRON /var/log/syslog | tail -10  
May 15 15:27:01 filter CRON[1326]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:28:01 filter CRON[1412]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:29:01 filter CRON[1488]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:30:01 filter CRON[1510]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:31:01 filter CRON[1518]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:32:01 filter CRON[1542]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:33:01 filter CRON[1559]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:34:01 filter CRON[1567]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:35:01 filter CRON[1589]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
May 15 15:36:01 filter CRON[1592]: (root) CMD (/usr/bin/python3 /home/usuario/edunetguard-worker/sync_pihole_from_db.py >> /home/usuario/edunetguard-worker/sync_pihole_from_db.log 2>&1)  
usuario@filter:~$
```

Veure el log del worker en temps real:

```
sudo tail -f /home/usuario/edunetguard-worker/sync_pihole_from_db.log
```

```
[INFO] Día 5 Hora 15:30:01  
[=] Sin cambios  
[INFO] Día 5 Hora 15:31:01  
[=] Sin cambios  
[INFO] Día 5 Hora 15:32:01  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)tiktok\.com$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)instagram\.com$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)cdninstagram\.com$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)x\.com$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)twitter\.com$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)twimg\.com$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)facebook\.com$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)fbcdn\.net$ -> 1  
[CAMBIO] EDUNET:REDES_SOCIAL | (\.|^)pinterest\.com$ -> 1  
[CAMBIO] EDUNET:FOROS_OCIO | (\.|^)forocoches\.com$ -> 1  
[OK] Pi-hole actualizado  
[INFO] Día 5 Hora 15:33:01  
[=] Sin cambios
```

### Consultar l'estat de les categories EDUNET a Pi-hole:

```
sudo sqlite3 /etc/pihole/gravity.db "SELECT comment, type, enabled FROM domainlist WHERE comment LIKE 'EDUNET:%' ORDER BY comment;"
```

```
EDUNET:JUEGOS|3|0
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:REDES_SOCIAL|3|1
EDUNET:STREAMING|3|0
EDUNET:STREAMING|3|0
EDUNET:STREAMING|3|0
EDUNET:STREAMING|3|0
EDUNET:STREAMING|3|0
```

### Veure usuaris i hashes de contrasenya a MariaDB:

```
mariadb -u edunetguard -p edunetguard -e "SELECT id, usuario, es_admin, activo, LEFT(password,7) AS hash_prefix FROM profesores ORDER BY id;"
```

```
MariaDB [edunetguard]> SELECT id, usuario, es_admin, activo, LEFT(password,7) AS hash_prefix FROM profesores ORDER BY id;
+-----+-----+-----+-----+-----+
| id | usuario | es_admin | activo | hash_prefix |
+-----+-----+-----+-----+-----+
| 1 | jmimo | 0 | 1 | $2y$10$ |
| 2 | dgarcia | 0 | 1 | $2y$10$ |
| 3 | otorrente | 0 | 1 | $2y$10$ |
| 4 | dmartinez | 0 | 1 | $2y$10$ |
| 5 | imartinez | 0 | 1 | $2y$10$ |
| 6 | admin | 1 | 1 | $2y$10$ |
+-----+-----+-----+-----+-----+
6 rows in set (0,001 sec)
```

### Veure permisos per assignatura:

```
mariadb -u edunetguard -p edunetguard -e "SELECT pa.profesor_id, p.usuario, pa.asignatura_id FROM profesor_asignatura pa JOIN profesores p ON pa.profesor_id=p.id ORDER BY pa.profesor_id;"
```

```
MariaDB [edunetguard]> SELECT pa.profesor_id, p.usuario, pa.asignatura_id FROM profesor_asignatura pa JOIN profesores p ON pa.profesor_id=p.id ORDER BY pa.profesor_id;
+-----+-----+-----+
| profesor_id | usuario | asignatura_id |
+-----+-----+-----+
| 1 | jmimo | 0377 |
| 1 | jmimo | 0379 |
| 2 | dgarcia | 0375 |
| 2 | dgarcia | 0379 |
| 3 | otorrente | 0374 |
| 3 | otorrente | 0378 |
| 4 | dmartinez | 0376 |
| 4 | dmartinez | TUT |
| 5 | imartinez | 1710 |
+-----+-----+-----+
9 rows in set (0,001 sec)
```

## Annex E. Manual breu d'ús

### Ús per al professorat:

1. Obrir el navegador i accedir a <http://10.0.0.51:8080>
2. Iniciar sessió amb usuari i contrasenya.
3. Seleccionar el grup (ASIX\_A o ASIX\_B) o accedir a l'horari personal.
4. Fer clic sobre una franja de l'horari.
5. Activar categories, afegir dominis extra o definir excepcions permeses.
6. Prémer «Guardar». El worker aplicarà els canvis automàticament en menys d'un minut.

### Ús per a l'administrador:

1. Iniciar sessió amb l'usuari administrador.
2. Accedir al «Panell administrador» des del menú principal.
3. Crear o modificar professors: nom, usuari, contrasenya i rol.
4. Activar o desactivar comptes segons sigui necessari.
5. Assignar permisos per assignatura a cada professor.
6. Verificar que l'horari personal del professor reflecteix els permisos assignats.

## Perspectiva de l'alumne

Des del punt de vista de l'alumne, el filtratge és transparent i no requereix cap configuració al seu equip. Quan un bloqueig és actiu, les consultes DNS dels dominis restringits retornen un error de resolució o mostren la pàgina de bloqueig de Pi-hole. L'alumne no pot evitar el filtratge canviant manualment el servidor DNS del seu dispositiu, ja que pfSense redirigeix tot el trànsit del port 53 cap a Pi-hole independentment del DNS que tingui configurat.

## Annex F. Estat honest del projecte

Estat	Elements
Confirmat i funcionant	Infraestructura virtual completa, LAN_EDU, LAN_PROFES, DNS forçat a Pi-hole, NAT Outbound, web al port 8080, Pi-hole al port 80, MariaDB, worker Python, cron cada minut, panell administrador, gestió de bloquejos, categories EDUNET, dominis extra i excepcions permeses.
Pendent o futur	HTTPS, còpies de seguretat automàtiques, mitigació avançada de DNS over HTTPS, protecció CSRF, auditoria d'accions administratives, importació d'horaris CSV, desplegament amb VLANs físiques en un centre real. Reset States de pfSense (funcionament parcial per SSH/pfctl). Hardening complet del sistema.

Font: elaboració pròpia.

## Annex G. Categories i dominis de bloqueig

Llista completa de categories i dominis sembrats a Pi-hole com a entrades EDUNET. Cada domini es desa com a expressió regular per bloquejar el domini principal i tots els subdominis.

Categoria	Dominis inclosos
EDUNET:IA	chatgpt.com, openai.com, claude.ai, anthropic.com, gemini.google.com, copilot.microsoft.com, perplexity.ai, meta.ai, deepseek.com, chat.deepseek.com, grok.com, x.ai, poe.com, mistral.ai, chat.mistral.ai
EDUNET:JUEGOS	roblox.com, rbxcdn.com, steamcommunity.com, steampowered.com, steamstatic.com, friv.com, crazygames.com, poki.com, miniclip.com
EDUNET:REDES_SOCIAL	tiktok.com, instagram.com, cdninstagram.com, x.com, twitter.com, twimg.com, facebook.com, fbcdn.net, pinterest.com
EDUNET:STREAMING	youtube.com,youtu.be,googlevideo.com,ytimg.com,youtubei.googleapis.com,twitch.tv,ttvnw.net,jtvnw.net,netflix.com,nflxvideo.net,nflximg.net,nflxso.net,spotify.com,scdn.co
EDUNET:DESCARGAS	mega.nz, mediafire.com, uptodown.com, softonic.com, 1fichier.com, gofile.io
EDUNET:COMPRA S_ONLINE	amazon.es, amazon.com, aliexpress.com, shein.com, temu.com, ebay.es, ebay.com
EDUNET:FOROS_OCIO	reddit.com, redd.it, 9gag.com, forocoches.com

Font: elaboració pròpia. Format de regex aplicat:  $(\.|^)\text{domini}\backslash.\text{ext}\$$

## Annex H. Relació entre objectius, proves i estat

Objectiu	Prova o evidència	Estat
Crear laboratori virtual	Captures de VirtualBox i configuració d'interfícies a pfSense	Assolit
Configurar pfSense	Captures d'interfícies, DHCP, NAT Outbound i DNS redirect	Assolit
Filtratge DNS per als alumnes	Consulta Pi-hole i prova DNS des del client alumne	Assolit
Separar xarxa de professorat	Proves de ping/curl des de LAN_PROFES i accés a la web	Assolit
Desenvolupar web i backend	Login, horari, gestió de bloquejos i panell administrador funcionant	Assolit
Gestionar permisos per assignatura	Consulta profesor_asignatura i horari personal filtrat	Assolit
Automatitzar amb worker i cron	Crontab, log del worker i consulta gravity.db	Assolit
Dominis extra i excepcions permeses	Prova de configuració i verificació de l'aplicació pel worker	Assolit
Analitzar Reset States	Proves SSH/pfctl des de srv-filter a pfSense	Parcial
Documentar proves i limitacions	Memòria i annexos tancats	Assolit

Font: elaboració pròpia.

## Nota sobre l'ús d'intel·ligència artificial

Durant l'elaboració d'aquesta memòria s'han utilitzat eines d'intel·ligència artificial com a suport en diverses fases del procés de redacció: per estructurar i organitzar els apartats, millorar la fluïdesa i la claredat del text, i generar esborranys inicials que posteriorment han estat revisats, adaptats i completats amb el criteri propi dels autors.

En l'àmbit tècnic, la IA també s'ha fet servir com a eina de suport durant el desenvolupament del projecte: per resoldre dubtes puntuals de codi, depurar errors en els scripts Python i el backend PHP, i buscar alternatives quan algunes decisions inicials no funcionaven com s'esperava.

En tot moment han estat els autors qui han pres les decisions tècniques, definit el contingut de cada secció i verificat que allò escrit reflecteix fidelment el treball realitzat. Les decisions d'arquitectura, la configuració de la infraestructura, el disseny de la base de dades i les proves de validació són fruit del treball propi. La IA ha actuat com a eina de suport, no com a substitut del criteri dels autors.

**Els autors assumeixen la responsabilitat íntegra sobre el contingut d'aquest document.**