
This is the **published version** of the bachelor thesis:

Gómez González, Ivan; Martí Escalé, Ramon, dir. Capacitant als estudiants : Emulació de xarxes en qualsevol lloc i en tot moment. 2024. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/290103>

under the terms of the  license

Capacitant als estudiants: Emulació de xarxes en qualsevol lloc i en tot moment

Ivan Gómez González

8 de febrer de 2024

Resum– El treball de fi de grau ha consistit en elaborar Nethermir, una eina de gestió i automatització de servidors i màquines virtuals amb una plataforma Web per ajudar en la docència de les assignatures de xarxes de telecomunicacions, en la que l'alumnat pot crear i connectar-se mitjançant un túnel VPN a les seves màquines virtuals per a la creació d'entorns de xarxa virtualitzats i el professorat pot gestionar l'estat global dels servidors. Aquest treball forma part de TECTONIC, un projecte d'innovació docent a les assignatures de xarxes de telecomunicacions que planteja una remodelació el sistema actual d'ensenyament als laboratoris. Una vegada implementades totes les funcionalitats, s'ha posat en producció el projecte, provant el sistema en un entorn real amb l'alumnat de Fonaments de Xarxes amb un resultat final satisfactori.

Paraules clau– Innovació docent, virtualització de xarxes, administració de servidors, desenvolupament web, automatització.

Abstract– The end-of-degree project has consisted of developing Nethermir, an automated server and virtual machine management tool with a Web platform to help teaching the subjects of telecommunications networks, in which students can create and connect through a VPN tunnel to their virtual machines to create virtualized network environments and professors can manage the global state of the servers. This work is part of TECTONIC, an educational innovation project in telecommunications networks subjects that aims to remodel the current system of teaching at the labs. Once all the functionalities have been implemented, the project has been put into production, testing the system in a real environment with the students of Foundations of Networks with a satisfactory final result.

Keywords– Teaching innovation, network virtualization, server administration, web development, automation.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

PER tal d'oferir als estudiants un aprenentatge sobre el funcionament de les xarxes, és necessari complementar els coneixements teòrics assolits a classe amb les pràctiques de laboratori, on els estudiants es puguin afrontar a les xarxes reals, configurant i gestionant-les de forma guiada.

El sistema d'aprenentatge fins a l'actualitat dels estudis d'Enginyeria de Telecomunicacions de la Universitat Autònoma de Barcelona s'ha basat en pràctiques físiques limitades a uns dissenys de xarxes molt simples, que escalen a sistemes de major complexitat amb l'ajuda de simuladors de xarxes que executen els alumnes a les seves màquines.

El model plantejat, però, no només incapacita als estudiants de treballar col·laborativament en els projectes virtuals, sinó que no permet la interconnexió entre els diversos grups de treball.

La solució a aquests problemes esmentats és TECTONIC (Transformació de la mEtodologia doCent dels laboraTORis

de telemàtica del grau d'engiNyeria de telecomunICació), un projecte d'innovació docent a les assignatures de xarxes de telecomunicacions que planteja remodelar el sistema actual d'ensenyament al laboratori mitjançant una plataforma on els estudiants puguin crear i gestionar xarxes virtualitzades de complexitat d'una forma senzilla des de qualsevol lloc.

Donada la magnitud del sistema i tot el projecte d'innovació docent com a base, al punt de partida es compta amb tota la infraestructura física del sistema ja implementada (Fig. 1) pel professorat d'Enginyeria de Telecomunicacions. Aquesta inclou: els servidors (x3 HP Proliant DL360 Gen8), que ja han sigut connectats i configurats; les màquines virtuals, que ja tenen la seva plataforma d'administració; les imatges corresponents, les interconnexions i les configuracions bàsiques de l'encaminador (Mikrotik RB4011). També existeix un mapping entre la VPN de l'alumne i la xarxa de la màquina mitjançant VRF-lite, que permet la coexistència de diversos encaminadors virtualit-

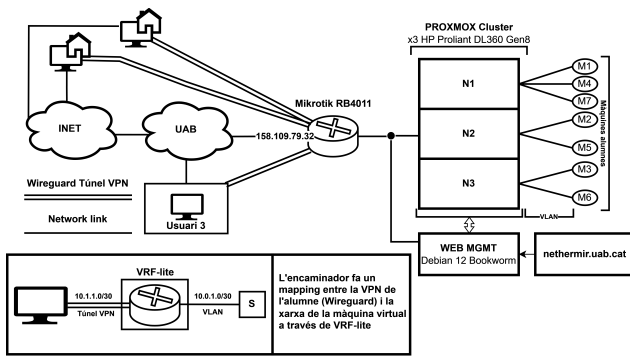


Fig. 1: Esquema de l'arquitectura física del projecte

zats, aïllant als estudiants i reaprofitant les adreces.

El treball de fi de grau planteja desenvolupar Nethermir, una plataforma de gestió i automatització de servidors per a TECTONIC, que permetrà als estudiants crear entorns de virtualització als quals podran accedir des de qualsevol lloc mitjançant un túnel VPN. Dins d'aquests entorns de virtualització, els estudiants muntaran xarxes virtualitzades i aquí sí, es connectaran les màquines virtuals en pont, permetent la interconnexió entre els diversos grups de treball.

Aquest treball compta amb un panell de control Web [1], on es permetrà al professorat crear imatges d'entorns de laboratori, modificar la base de dades (crear assignatures i vincular plantilles, eliminar estudiants / grups) i monitorar i modificar l'estat de les màquines virtuals que farà servir l'alumnat a les seves pràctiques.

Per part de l'alumnat Nethermir permetrà als estudiants mitjançant la plataforma web registrar-se, iniciar sessió i generar el seu entorn de virtualització. En generar-se la seva màquina basada en la plantilla de l'assignatura, els alumnes es podran connectar remotament aquesta màquina virtual des de qualsevol part mitjançant sessions gràcies a un sistema de túnels VPN automàtic.

2 OBJECTIUS

Per tal de definir correctament el desenvolupament d'una aplicació on les noves funcionalitats poden causar que aquest esdevingui un procés infinit, serà necessari que es proposin correctament els nostres objectius principals.

Separarem el desenvolupament en tres objectius amb la mateixa prioritat, cadascuna amb els seus subobjectius, els quals es llistaran per ordre de prioritat, de major a menor:

Desplegament d'una interfície Web: Mitjançant aquesta interfície, tot usuari, sigui professorat o estudiant, podrà accedir al seu portal de control, on tindrà accés a gestionar les seves màquines virtuals. El professorat, a més, podrà gestionar també des d'aquesta interfície la base de dades d'alumnes, grups i assignatures. A continuació s'esmenten els subobjectius plantejats:

- Disseny d'un formulari de registre i inici de sessió
- Disseny d'un panell de control pel professorat
- Disseny d'un panell de control per l'estudiant
- Desenvolupament d'un gestor de galetes de sessió.

Implementació d'una API REST: Per tal de poder comunicar la interfície Web amb el backend, necessitarem

desenvolupar una API que utilitzarem com a pont de comunicació entre el frontend i els diversos mòduls del backend.

Disseny i desenvolupament del gestor de servidors: El backend serà el nucli del nostre gestor de servidors, i serà l'encarregat de, donats els seus paràmetres i les crides rebudes per l'API, donar un correcte funcionament del gestor. A continuació s'esmenten els subobjectius plantejats:

- Disseny i implementació una base de dades
- Implementació un administrador de màquines virtuals
- Implementació un script d'autoconfiguració de l'encaminador
- Connexió amb el servidor SMTP de la UAB
- Desenvolupament d'un gestor de galetes de sessió
- Aplicació d'un logger rotatiu personalitzat.

3 ESTAT DE L'ART

En aquest apartat es comenten les solucions més populars pels diferents mòduls del projecte per tal de poder escollir de forma correcta quines utilitzarem a la implementació. S'esmenten les tecnologies més populars per la interfície Web, la base de dades, l'administrador de màquines virtuals, l'autoconfigurador de l'encaminador, el servidor de correus i el logger rotatiu personalitzat.

3.1 Plataformes de docència

Actualment, existeixen implementacions similars a aquest projecte a l'entorn acadèmic, que intenten simular un internet mitjançant l'administració d'encaminadors, commutadors, amfitrions i sistemes autònoms per part dels estudiants [2]. L'exemple més notable, segurament és el projecte mini-Internet [3] realitzat per ETH Zurich. Aquest, permet ensenyar de forma interactiva tots els conceptes teòrics apresos sobre administració de xarxes, en entorns que tot i ficticis, intenten simular el que ben podrien ser entorns reals amb totes les seves complexitats. La cooperació entre els estudiants per tal de formar l'Internet, dona un gran incentiu als estudiants no només en aprendre, sinó en aprofundir i indagar en els ensenyaments impartits a classe.

Mentre que aquesta solució se centra únicament en el desenvolupament d'un entorn virtual que simula la internet, el projecte que plantegem, automatitza totalment el procés de posada en marxa i la gestió i monitoratge dels diferents sistemes autònoms, mitjançant la gestió de les màquines virtuals per l'API de l'hipervisor utilitzat al projecte. A la vegada, ofereix un sistema totalment user-friendly amb la seva interfície Web, que permet una ràpida adaptació dels estudiants al model d'ensenyament. No només això, sinó que és un projecte modular que permet l'adaptació tant de noves funcionalitats com les modificacions en el seu funcionament per tal d'adequar-se a les necessitats de cada assignatura.

3.2 Interfície Web

Per al desenvolupament d'interfícies Web és molt habitual utilitzar eines que treballin per sobre HTML, CSS i Javascript, que permeten crear interfícies Web de forma ràpida,

esquemàtica i amb flexibilitat. Aquest seria el cas de React [4], Angular [5] i Vue [6].

React és l'opció més popular, i per tant compta amb una major comunitat, ecosistema, llibreries i contingut de referència. Angular és l'opció més completa, ja que no requereix tantes llibreries externes, però té una corba d'aprenentatge major i ofereix un desenvolupament i estructura tancada, que pot ser de gran utilitat en projectes empresarials. Per últim, Vue, també consta amb un gran nombre de llibreries externes, i tot i que la seva popularitat és una mica menor a React, aquest consta d'una corba d'aprenentatge menor a les altres dues alternatives.

3.3 Base de dades

A l'hora d'escollir quina base de dades utilitzar, la primera decisió és si optar per una base de dades relacional (RDB), que permet establir relacions entre les diverses entitats registrades, o una base de dades no relacional (noSQL), que no emmagatzema la informació en taules, que permet una major flexibilitat, escalabilitat i emmagatzematge. En una base de dades relacional, s'assegura la integritat de les dades a canvi d'una major dificultat d'escalabilitat. Per contra, una base de dades noSQL, ofereix una major escalabilitat, a canvi de la consistència i una corba d'aprenentatge major.

3.4 Administrador de màquines virtuals

Per tal d'administrar les màquines virtuals de forma nativa, haurem d'escollir quin hipervisor de tipus 1 utilitzar. Algunes de les opcions més populars són les següents:

- VMware ESXi [7]: Una solució per a empreses amb serveis de gran escala, amb un gran suport tècnic per part de VMware i un gran cost d'entrada.
- Xen [8]: Una solució de codi obert desenvolupada per la Fundació Linux. Ofereix una gran modularitat, permetent una millor escalabilitat. El servei de suport i documentació en línia és de gran qualitat.
- Proxmox [9]: Una altra solució de codi obert, ofereix moltes de les característiques d'ESXi de forma gratuïta. Manté suport per una gran varietat de sistemes d'emmagatzematge (GlusterFS, NFS, LVM...).

Una vegada escollida l'eina per administrar les màquines virtuals, haurem d'escollir com comunicar-nos amb aquesta. Tenim dues opcions:

- L'API que ens proporcioni la tecnologia escollida, oferint un control total sobre el desenvolupament, assegurant que no utilitzem eines obsoletes.
- Una llibreria de client creada per la comunitat, que facilita la implantació fent transparent la complexitat de la connexió, autenticació i semàntica de l'API. És possible que aquesta opció estigui incompleta o tingui documentació nul·la.

3.5 Autoconfigurador de l'encaminador

Per tal de comunicar-nos amb l'encaminador muntat pel professorat, igual que amb el cas anterior, tenim dues possibilitats; utilitzar l'API que ens proporciona Mikrotik oficialment, o una llibreria de client.

El tret característic de totes dues opcions en aquest cas, és que cap ofereix una serialització de les connexions, i l'encaminador només permet una connexió simultània. Aquest fet, en cap altre entorn de desenvolupament, no hauria suposat cap problema, però el nostre entorn és node.js, un sistema totalment asíncron, on si es vol serialitzar un procés, aquest s'ha d'externalitzar d'alguna manera.

Per tal d'externalitzar aquests processos, la tècnica més utilitzada és utilitzar una llibreria externa, que es connecta amb una base de dades noSQL per tal de gestionar cues de treball, permetent aplicar prioritats, limitadors d'amplada de banda, persistència i sovint interfícies Webs per a la gestió d'aquestes cues i l'anàlisi del seu rendiment.

Les llibreries més populars són les següents:

- Bull [10]: És la llibreria més popular. Ofereix el nombre més gran de funcionalitats amb un rendiment de qualitat, obté actualitzacions constants i té una gran comunitat que manté el desenvolupament del projecte
- Kue [11]: Planteja una solució més senzilla que Bull, amb un nombre menor de funcionalitats però reduint la complexitat del producte pels desenvolupadors.
- Bee [12]: És l'opció més recent i està oferint uns grans resultats. Elimina gran quantitat de funcionalitats, permetent oferir un servei molt més ràpid [13] que les seves alternatives.

3.6 Servidor de correus

Per tal d'oferir una solució per a l'enviament de correus, les opcions són limitades, ja que gairebé tota la comunitat de node.js utilitza la mateixa llibreria per externalitzar tots els processos de connexió amb els servidors SMTP. Aquesta llibreria és Nodemailer [14], que ofereix una solució senzilla però amb un gran nombre de funcionalitats.

3.7 Logger rotatiu

Un logger rotatiu ens permet emmagatzemar diàriament de forma comprimida l'estat del nostre sistema, permetent així una posterior anàlisi de l'estat del sistema en un moment concret en el temps. Depenent de la magnitud del nostre projecte, el sistema de logging que escollirem tant per al desenvolupament com per al monitoratge i traçabilitat de bugs, serà una de les tres opcions següents, ordenades per ordre de complexitat de menor a major:

- Console [15]: Logger per defecte ofert per node.js, ofereix les funcionalitats més bàsiques i pot ser d'utilitat per a projectes petits.
- Pino [16]: En sistemes on el rendiment és la qualitat més desitjada Pino ofereix una alternativa simple però molt ràpida.
- Winston [17]: La llibreria externa més popular per node.js, ofereix un sistema de logging en .json amb marques de temps i nivells que ofereix un monitoratge i traçabilitat de qualitat.

4 METODOLOGIA

Per al desenvolupament d'aquest projecte, la metodologia escollida ha sigut la d'un procés àgil [20]. Aquest, es basa en els principis d'un desenvolupament iteratiu, incremental i evolutiu, que permet obtenir de forma ràpida feedback per al desenvolupador, per tal de minimitzar el risc i assegurar que el producte final s'adequa correctament a les necessitats dels usuaris. Aquesta metodologia ens permet dividir un desenvolupament de llarga durada, en tantes subtasques com iteracions tingui el projecte, aconseguint així una major eficiència a l'hora d'implementar codi.

En aquest cas, les iteracions han sigut d'una a dues setmanes, on s'han organitzat els divendres reunions amb les parts interessades (en aquest cas, amb el professorat a càrrec del projecte d'innovació docent), per tal d'avaluar els avenços realitzats, i decidir si es necessita un replanteig de la solució aportada, o es pot continuar favorablement amb el disseny i implementació de la següent funcionalitat.

És important, que una vegada dut a terme l'anàlisi i disseny del sistema, llancem ràpidament un Minimum Viable Product (MVP) del controlador Web per tal d'obtenir feedback el més ràpid possible.

5 PLANIFICACIÓ

Passem amb la planificació del projecte, on donada la metodologia àgil esmentada amb principis incrementals i evolutius, podem ja definir l'ordre en el qual es desenvoluparan les tasques. Per tal de donar una cohesió a l'informe, aprofitarem el llistat d'objectius, per tal de definir les tasques a desenvolupar.

Com els tres objectius principals que hem marcat tenen la mateixa prioritat, anirem desenvolupant de forma paral·lela tots tres, aconseguint donar una visió al client dels avenços portats a cap en tot moment.

Per planificar el desenvolupament, s'ha creat un diagrama de Gantt (Fig. 6, Apèndix), que permet mostrar de forma visual l'organització setmanal del projecte i les seves subtasques.

6 DESENVOLUPAMENT DEL PROJECTE

Explicarem en detall l'anàlisi dels requisits, el disseny i les tecnologies escollides d'entre les esmentades a l'estat de l'art i les funcions implementades de cadascun dels apartats de la planificació: Arquitectura general, API REST, base de dades, administrador de màquines virtuals, autoconfigurador per l'encaminador, servidor de correus, logger rotatiu personalitzat, interfície Web, inici de sessió, registre, panell de control de l'estudiant, panell de control del professorat i testing.

6.1 Arquitectura general

En aquest apartat es planteja l'arquitectura general del projecte, aquesta inclou l'esquema general del seu disseny i la diferenciació entre els seus tres components fonamentals, el Frontend, el Backend, i l'API d'interconnexió.

6.1.1 Anàlisi

Si volem assegurar la cohesió, el manteniment, la reusabilitat i el testeig del nostre programa necessitem establir dos requisits imprescindibles: el desacoblament i el principi de responsabilitat única. Amb el desacoblament, cada mòdul pot treballar de forma totalment autònoma i modificar una tasca no repercuteix a la resta del projecte. D'altra banda, el principi de responsabilitat única (SRP) planteja que cada mòdul ha de tenir una única funcionalitat. Ambdós factors milloren aquests aspectes, ja que es compta amb un únic punt de testeig que permet una fàcil comprensió i elaboració del codi.

Com a entorn de programació Web per al desenvolupament de tot el projecte, s'opta per node.js. Donada la seva popularitat, disposa d'un gran suport, comunitat, rendiment i estabilitat. Pel control de versions s'ha optat per l'eina més estandarditzada, GitHub, que, mitjançant Git, permet oferir una solució al núvol amb control d'accés i integració continua.

6.1.2 Disseny

Nethermir consta d'un gran nombre de mòduls interconnectats, així que serà necessari establir el disseny del projecte global per a la seva anàlisi.

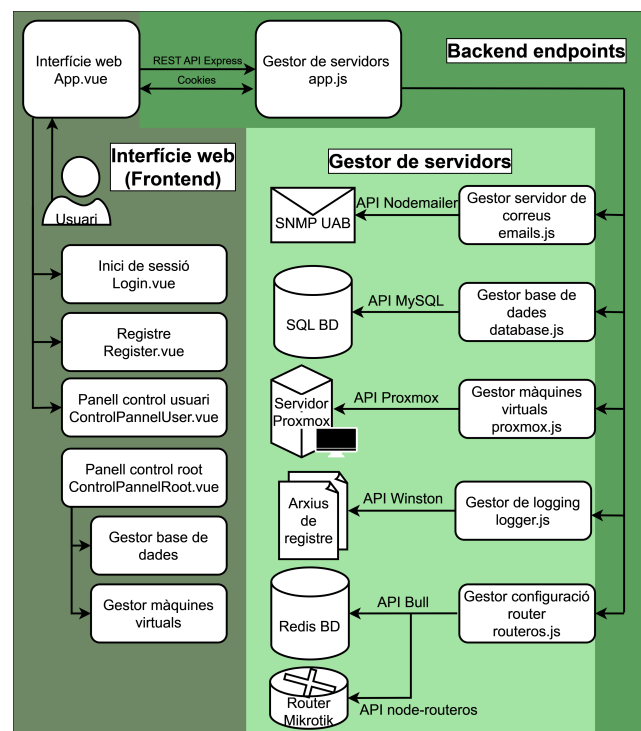


Fig. 2: Esquema del disseny del projecte

A la Fig. 2 es pot veure l'esquema del disseny del projecte, amb els tres components fonamentals: El frontend a la part esquerra, el backend a la part dreta, i la interconnexió entre els dos mitjançant els endpoints de l'API.

Un tret important d'aquest model, és que no representa referències circulars, o cap altre tipus de connexió entre els diferents submòduls entre si. Aquesta comunicació en arbre, permet els dos requisits esmentats a l'anàlisi de requisits: el desacoblament i el principi de responsabilitat única.

6.2 API REST

Com es pot observar a l'esquema del disseny global del projecte, l'API REST és la que es comunica tant amb el Frontend, com amb tots els mòduls del Backend, i per tant és qui conté tota la lògica de les funcions globals, atorgant al projecte d'un desacoblament total entre els dos sistemes i un únic punt de testeig.

6.2.1 Anàlisi

Si volem estructurar les crides als endpoints d'una forma sistemàtica, haurem d'utilitzar un protocol que ens permeti definir de forma estricta la sintaxi d'aquestes. La solució més estandarditzada és l'ús d'API REST, on existeixen quatre models de crides: GET (recuperar informació), PUT (inserir informació), POST (modificar informació) i DELETE (eliminar informació).

Per desenvolupar l'API REST, farem servir Express [18], un framework específicament per aquesta tasca, amb molt poca configuració necessària i amb la capacitat d'aplicar el protocol REST de forma nativa.

6.2.2 Disseny i implementació

Per assegurar que les crides a les funcions només són fetes pels usuaris autoritzats, s'implementa un servei de galetes de sessió que es generen en iniciar sessió i duren una hora. Al principi de l'execució de cada endpoint, comprovarà que aquesta galeta estigui i no hagi caducat. Si ha caducat, eliminarem la galeta de l'usuari i rebutjarem la petició. Una vegada definits els quatre models de crides oferts per les API REST, podem separar els nostres endpoints de la següent forma:

- GET: statusAllVMs, statusVM, startingTimeVM, groups, emails, subjects, login, checkCookie
- PUT: register, machine, subject
- POST: activateSubject, deactivateSubject, activateMachine, stopMachine, resumeMachine, suspendMachine
- DELETE: group, email, subject, machine, cookie, restartDatabase

Per mostrar el funcionament dels endpoints i com contenen la lògica de totes les funcions, esmentarem a alt nivells els passos d'execució quan s'envia una petició DELETE /group des del frontend.

1. Eliminar el grup de la base de dades (database.js)
2. Eliminar la màquina virtual assignada al grup (prox-mox.js)
3. Eliminar la configuració del router assignada al grup (routeros.js)
4. Enviar un codi 200 OK / 400 ERROR. (app.js)

Com es pot observar se segueix el model plantejat a l'esquema del disseny global. Tota la lògica queda encapsulada a l'endpoint, i als mòduls auxiliars només es criden funcions atòmiques amb un propòsit únic.

6.3 Base de dades

La base de dades a implementar ens permetrà emmagatzemar de forma no volàtil tota la informació necessària sobre el projecte, incloent-hi la gestió d'assignatures, grups i emails, i oferint una connexió amb la interfície Web per a una fàcil gestió i anàlisi de l'estat del sistema.

6.3.1 Anàlisi

Esmentades les dues opcions per implementar la base de dades a l'estat de l'art (relacional / no relacional), s'escull treballar amb una base de dades relacional: MySQL [19], que permet una gestió senzilla, sense cost, i amb una filosofia codi obert. S'ha escollit una base de dades relacional, ja que el disseny serà d'una complexitat molt limitada i no té com a objectiu treballar amb grans volums de dades. Una vegada establerta la nostra eina de treball, haurem d'analitzar quins són els components que volem emmagatzemar. Aquests seran principalment tres:

- Assignatures: El professorat haurà de poder crear, eliminar i modificar les assignatures existents a les quals els alumnes es podran registrar.
- Grups: Serà necessari emmagatzemar informació sobre els estudiants pertanyents al grup, la màquina virtual que gestionen i l'assignatura a la qual pertanyen.
- Emails: Per cada estudiant, únicament enregistrarem el seu email.

Plantejats els requisits, podem passar al disseny de la base de dades.

6.3.2 Disseny

Una vegada plantejat l'anàlisi de requisits, podem crear una base de dades que compleixi amb les nostres necessitats. A la Fig. 3 es mostra el diagrama Entitat - Relació de la base de dades plantejada.

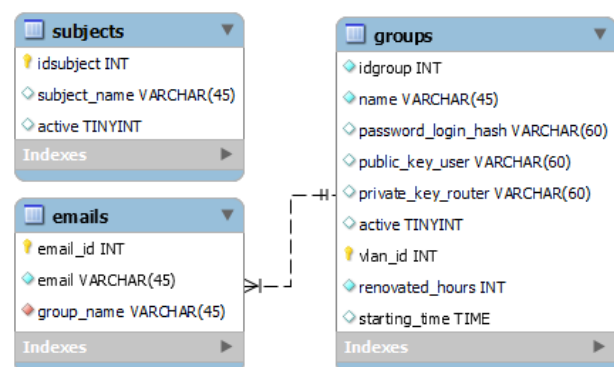


Fig. 3: Diagrama Entitat-Relació

En aquest esquema, es poden veure tots els requisits esmentats excepte la relació entre grups i assignatures. Aquest fet, permet el desacoblament de les dues taules, permetent la modificació per part del professorat de les assignatures sense cap mena de restricció.

Per aconseguir, per tant, emmagatzemar la informació sobre l'assignatura al grup, necessitem el camp "name", que

tindrà el següent format: Sigles Assignatura - Any - Semestre - ID Assignatura + ID Grup.

Per tant, el grup FX-2022-1-100, pertany a l'assignatura Fonaments de Xarxes (FX) de l'any 2022, primer semestre, l'ID de l'assignatura és l'1 (és la primera assignatura creada a la base de dades, s'assigna un nou ID cada semestre) i el ID del Grup és el 00 (Fins a 99 per assignatura).

A més, haurem d'emmagatzemar la contrasenya del grup encriptada amb un hash, i informació sobre la màquina virtual (hora d'encesa, nombre d'hores).

6.3.3 Implementació

Pel que fa a les funcions implementades per a la interacció amb els endpoints del backend, s'ha optat per continuar el model una implementació transparent i agnòstica, que permet el canvi a una altra base de dades, sense haver de modificar en cap mode els endpoints. D'aquesta forma tindrà les funcions atòmiques següents, agrupades de forma lògica segons els components que accedeixen o modifiquen:

Groups: getGroups, deleteGroup, registerGroup, authenticate, getGroupData, activateGroup

Emails: getEmails, deleteEmail, getEmailsFromGroupName

Assignatures: getSubjects, deleteSubject, addSubject, activateSubject, deactivateSubject

Màquines Virtuals: getStartingTimeVM, getRenovationHoursVM, changeRenovationHoursVM

Base de dades: restartDatabase

6.4 Administrador de màquines virtuals

Cada grup tindrà assignada una màquina virtual mitjançant la qual realitzarà les seves pràctiques de laboratori de les assignatures de xarxes. És necessari un sistema que s'encarregui d'administrar el control a aquestes màquines: La creació, el monitoratge i el control d'aquests sistemes haurà de ser gestionat de forma automàtica i permetre al professorat i estudiants interactuar amb aquest per crear, encendre i parlar les màquines quan ho necessitin. Una vegada l'assignatura termini, s'ha d'establir un sistema de neteja de màquines virtuals per poder alliberar recursos al servidor.

6.4.1 Anàlisi

Com que l'administrador de màquines virtuals ha sigut escollit i implementat pel professorat, no s'ha hagut de fer una anàlisi de les diferents opcions al mercat. Aquest ha estat Proxmox, un hipervisor de tipus 1, que permet una virtualització en mode natiu i s'instal·la directament sobre l'equip amfitrió. Aquest mètode, permet dedicar el servidor complet en exclusivitat a la virtualització de tot el sistema.

El disseny de l'entorn de Proxmox, consta actualment de tres servidors, on les màquines dels usuaris es distribueixen equitativament mitjançant un algorisme de round-robin

6.4.2 Disseny

Per obtenir a quin servidor es troba una màquina seguint un algorisme round-robin, necessitem un mètode ràpid, per tant, farem una operació de mòdul tres del grupID.

L'opció escollida per a comunicar-nos amb Proxmox, ha sigut una llibreria de client [20], donada la seva fàcil implementació i el fet que permeti que el codi de desenvolupament sigui més clar i concís.

6.4.3 Implementació

Pel que fa a les funcions implementades per a la gestió de màquines virtuals, continuem amb la filosofia de transparència. Agrupem les funcions de forma lògica segons si aquestes canvien l'estatus de la màquina, el consulten, o es clona / elimina un entorn virtualitzat:

Machine change status: activateMachine, stopMachine, resumeMachine, suspendMachine

Machine get status: getStatusVM, getStatusAllVMs, getActiveVM

Machine creation: cloneMachine, deleteMachine, modifyMachineVLAN

6.5 Autoconfigurador per l'encaminador

Per tal de connectar tant el programari de gestió com les màquines virtuals amb l'exterior, és necessari establir algun mètode per a comunicar-nos amb l'encaminador que actua com a tallafoc del sistema. Per tant, haurem de crear un codi per configurar automàticament l'encaminador.

6.5.1 Anàlisi

El codi a desenvolupar ha de ser capaç de generar configuracions, i eliminar aquelles que ja no siguin necessàries, a més d'un sistema de control per assegurar que la generació de les configuracions es fa de forma correcta.

S'opta per treballar amb la llibreria externa node-routeros [21], que, tot i que el projecte està discontinuat, ofereix la solució més completa i amb un funcionament i sintaxi molt similar a la que utilitza RouterOS, permetent així una configuració molt senzilla de l'encaminador.

Com s'ha esmentat a l'Estat de l'Art, també és necessari crear un sistema de cues extern per tal de gestionar la serialització de les connexions a l'encaminador. Esmentades totes les possibilitats en els apartats anteriors, s'escull treballar amb Bull, ja que essent el procés de configuració de l'encaminador de l'escala dels segons, no ens preocupa el rendiment que ens podria oferir Bee, però si ens poden utilitzar funcionalitats que ofereix Bull i no ofereix pas Bee com pot ser l'assignació de prioritat als treballs.

L'entorn escollit per a la gestió de cues, Bull, necessita que hi hagi implementada una base de dades Redis [22]. Per a una posada en marxa ràpida i dinàmica, s'opta per un entorn basat en containers Docker, que ens permet desplegar una base de dades Redis i obrir les connexions necessàries amb el Backend. Aquest, serà l'encarregat d'encuar totes les peticions que hi arribin a l'encaminador. Aquestes peticions seran de dos tipus, o de generació o d'eliminació.

Per últim, necessitem establir un sistema de túnels virtuals privats que permetrà als estudiants connectar-se a les seves màquines virtuals des de qualsevol lloc. Per a la creació de túnels VPN, es treballa amb WireGuard [23], que ha sigut plantejat com a requisit per part del professorat. Aquest programari funciona amb un sistema de clau pública, on haurem de crear un parell publicKey secretKey per a cada connexió entre grup i encaminador. Aquest parell de claus,

haurà de ser enviat a l'encaminador per tal de gestionar els túnels de la forma pertinent.

6.5.2 Disseny i implementació

El primer cop que un grup es registra, es generaren les configuracions de la VPN WireGuard i aquestes s'envien per correu als integrants del grup, però no s'aplicaran a l'encaminador ni es crearà les màquines virtuals fins que iniciïn sessió per primer cop (per tal d'evitar atacs de bots).

Una vegada el grup iniciï sessió per primer cop, s'enviarà la configuració a l'encaminador. Si aquest procés, per alguna raó, acaba de forma sobtada i el codi llença un error, s'eliminarà tota la configuració existent de l'encaminador sobre aquest grup per evitar es quedi codi zombi de grups inexistents. Si, per contra, no hi ha cap problema en la generació, aquesta configuració serà persistent fins que aquest grup sigui finalment eliminat, moment en el qual es cridarà a l'eliminació d'aquesta.

Les funcions atòmiques en aquest cas només seran dues: `generateRouterOSConfig` i `deleteRouterOSConfig`.

6.6 Servidor de correus

El servidor de correus integrat ens permet enviar de forma segura i automatitzada correus als estudiants. Comunicant el nostre sistema amb el servidor SMTP de la Universitat Autònoma de Barcelona, podem enviar-li les nostres peticions i aquest serà l'encarregat de l'enviament dels correus electrònics.

6.6.1 Anàlisi

Donat el consens en tota la comunitat que Nodemailer és el millor paquet per a la generació de correus electrònics, s'escull treballar amb aquest. Nodemailer, com ja s'ha comentat, s'ha de comunicar amb el servidor SMTP de la Universitat Autònoma de Barcelona.

Haurem d'enviar dos tipus de correus. Per una banda, quan els estudiants creen un grup, s'envia un correu amb les seves credencials d'inici de sessió, un PDF amb les instruccions d'ús i l'arxiu de configuració de la VPN WireGuard, que inclou la clau secreta del grup, la clau pública del servidor i la xarxa assignada al túnel. Això ens permet que, donat que només es poden registrar correus electrònics institucionals, evitem la generació massiva amb l'ús de bots de màquines virtuals que trencarien el sistema actual.

Per l'altra banda, s'ha d'enviar un correu als estudiants abans que la seva màquina sigui apagada per tal de permetre a aquests allargar la durada de la seva reserva.

En últim lloc, per evitar que les màquines virtuals es puguin quedar enceses de forma indefinida, s'ha d'establir un sistema de reserves amb enviament de correus.

6.6.2 Disseny

Analitzats els requisits del servidor de correus, a continuació s'explicarà el disseny del sistema de reserves implementat:

Els usuaris tenen la capacitat d'encendre la seva màquina virtual per un nombre d'una a quatre hores. Aquesta crida, encén un comptador, que, quan falti mitja hora per finalitzar

i sempre que no s'hagi allargat el temps de la reserva, avisarà de l'apagament proper de la màquina virtual i passada la mitja hora apagarà la màquina. Si, per contra, s'allarga la durada de la reserva, aquest procés encendrà un altre comptador amb la mateixa funcionalitat que l'anterior. Així s'assegura que les màquines són només apagades quan toca i sempre amb un avís previ.

6.6.3 Implementació

Donat el disseny plantejat, comptarem amb dues funcions atòmiques que cridarem des del nostre gestor de servidors en els casos esmentats amb anterioritat: `sendWarningMail` i `sendPasswordEmail`.

Tota la lògica referent a quan s'ha d'enviar els correus de warning, queda delegada a l'endpoint `POST activateMachine`. Per altra banda, es cridarà a la funció `sendPasswordEmail` dins l'endpoint `PUT register`.

6.7 Logger rotatiu personalitzat

El logger rotatiu ens permetrà generar arxius de logging on emmagatzemarem l'estat del nostre sistema. Al ser rotatiu, diàriament guardarem tota la informació generada en un arxiu comprimit per tal d'analitzar-lo en detall quan ho necessitem.

6.7.1 Anàlisi

Si es tractés del desenvolupament d'un producte que no compta d'un seguiment o d'una complexitat molt limitada, s'hauria optat per treballar amb el logger console ofert per `node.js`, però com un dels requisits del servei és que el logger compti amb un sistema de loggins rotatius que ens permeti monitorar de forma diària l'estat del nostre sistema per a una anàlisi de bugs ràpida, s'opta per treballar amb Winston, donat el seu gran nombre de funcionalitats ens assegura que complirà totes les nostres necessitats.

6.7.2 Disseny i Implementació

El disseny compta amb dos tipus de logs principalment; els d'informació, que serveixen per a monitorar l'estat del sistema en tot moment, i els d'error, que ens permeten veure en quines situacions el sistema funciona d'una forma inesperada i es produeix cap error. Es creen dos arxius, per tant, `nethermir.log`, on s'emmagatzema tots els missatges de logging de tots els nivells i `nethermir.err`, on únicament s'emmagatzema els missatges d'errors creats pel sistema.

Per altra banda, s'estableix un sistema rotatiu on diàriament, l'arxiu `nethermir.log` es crea i es neteja un arxiu amb la data pertinent per a una posterior anàlisi més entenedora.

6.8 Interfície Web

Mitjançant la interfície Web, tant els estudiants com el professorat faran totes les seves gestions necessàries. Es planteja un sistema senzill d'utilitzar, però que porti tota la informació i funcionalitats necessàries per a que no sigui necessari utilitzar cap altre sistema extern per a gestionar la base de dades, l'autoconfigurador o les màquines virtuals.

6.8.1 Anàlisi

Com a anàlisi de requisits generals de la interfície Web, serà necessari establir un selector d'idiomes i un mode clar/fosc per a facilitar una major accessibilitat a tothom. A més, s'haurà de seguir l'esquema amb accents de colors verds de la Universitat Autònoma de Barcelona. Haurà de constar també amb un disseny Web responsiu i adaptatiu per entorns mòbils (Fig. 7, Apèndix). Com a eina de desenvolupament per al frontend s'opta per Vue, que donada la familiaritat i experiència amb l'entorn, proporciona una major velocitat i facilitat a l'hora de desenvolupar el projecte. A més, aporta més flexibilitat que no pas altres alternatives com Angular, ja que s'adapta perfectament a les necessitats.

6.8.2 Disseny

El disseny és modular i mitjançant l'ús de SFC (single file components) es permet una major organització i esquematització, amb un codi molt més fàcil de llegir i amb la possibilitat de reutilitzar codi de forma còmoda. Es crea un arxiu .vue per cada "pantalla", i aquestes consten de tants components com sigui falta, abstraient tota la lògica a aquests.

6.9 Registre

El formulari de registre, serà el primer punt de contacte dels estudiants amb l'entorn Web, i permetrà a aquests escollir l'assignatura que cursen i els integrants del grup. A tots aquests, se'ls assignarà unes credencials d'inici de sessió que seran enviades als correus facilitats pels alumnes al formulari.

6.9.1 Anàlisi

Pel que fa al formulari de registre, compta amb un nombre major de camps. Aquests són, un selector de curs / Assignatura que mostri només les assignatures activades pel professorat, el nombre d'integrants del grup, tants camps per introduir emails com integrants tingui el grup, i un botó per enviar el formulari a més del control d'errors pertinent.

6.9.2 Disseny i implementació

Com a imatge de fons tenim un disseny gràfic de les Columnes de Bellaterra, escultura distintiva de la universitat, extreta de la identitat corporativa de la universitat. El formulari de registre (Fig. 4) implementat amb Vue compta amb tots els camps requerits a l'anàlisi.

6.10 Inici de sessió

A aquest panell, els estudiants podran iniciar sessió mitjançant les credencials que han sigut enviades al correu amb el qual han registrat el grup, com s'ha esmentat en apartats anteriors.

6.10.1 Anàlisi

Són necessaris només quatre elements principals: introduir usuari, introduir contrasenya, un botó per mostrar/ocultar la contrasenya i un altre per enviar la nostra sol·licitud. Si en introduir les dades tenim algun error, s'ha de comunicar visualment aquesta informació.

Fig. 4: Formulari de registre, amb el mode clar i llenguatge anglès. S'intenta registrar sense emplenar tots els emails.

6.10.2 Disseny i implementació

De nou, com a imatge de fons, tenim un disseny gràfic de les Columnes de Bellaterra. El formulari d'inici de sessió dissenyat amb Vue (Fig. 8, Apèndix) compta amb tots els camps requerits a l'anàlisi.

6.11 Panell de control de l'estudiant

Una vegada els estudiants d'un grup iniciïn sessió, se'ls mostrarà el seu panell de control, on podran crear els seus entorns virtualitzats i una vegada generats, interactuar amb ells, fent reserves de temps de les màquines i monitorant l'estat de les seves màquines.

6.11.1 Anàlisi

L'experiència dels estudiants, ha de ser totalment guiada i senzilla. Quan inicien sessió per primer cop a la plataforma, hi haurà un procés de seguiment amb l'estudiant, que el guiarà en el seu procés de creació de la seva màquina virtual. El procés és totalment transparent i automàtic i, per tant, es necessita molt poca intervenció de l'estudiant. Una vegada acabat el procés, l'estudiant podrà encendre la màquina x hores, estendre aquesta durada, posar en pausa la màquina i parar la màquina. L'estat actual de la màquina ha de ser visible a l'estudiant.

6.11.2 Disseny i implementació

A les següents figures es pot veure el disseny i la implementació final, que consta primerament d'un únic botó per a la creació de la màquina virtual (Fig. 9, Apèndix) per aconseguir aquesta simplicitat i poca intervenció de l'estudiant.

Mentre es clona la màquina virtual (Fig. 10, Apèndix), es mostra un missatge d'espera.

Una vegada acabada (Fig. 11, Apèndix), es mostra finalment el monitoratge de la màquina virtual amb les seves opcions de control.

6.12 Panell de control del professorat

Per altra banda, si en iniciar sessió s'utilitzen les credencials assignades al professorat, es passa a un panell de control

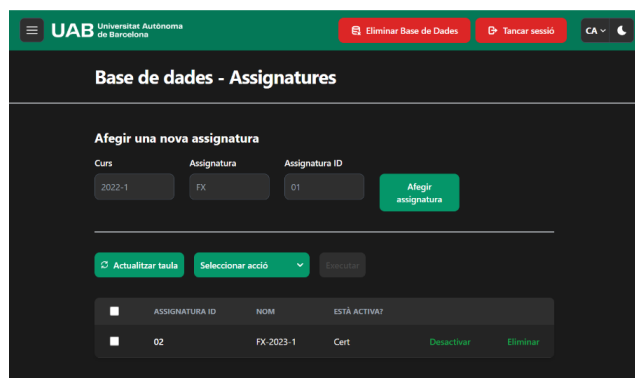


Fig. 5: Panell de control del professorat, amb el menú lateral tancat.

d'administració, on tenim una visió i control general de totes les taules de la base de dades (assignatures, grups i estudiants), i el monitoratge i gestió de les màquines virtuals de tots els estudiants.

6.12.1 Anàlisi

El panell de control del professorat, tot i que senzill d'utilitzar, haurà de contenir totes les funcionalitats necessàries per a administrar el sistema. Per tant, haurà de tenir un control total sobre els entorns de virtualització i la base de dades (gestió de grups, estudiants, i creació d'assignatures)

6.12.2 Disseny i implementació

Començant pel menú, consta amb una capçalera on podrem ràpidament reiniciar l'estat actual de la base de dades (Fig. 12, Apèndix) per fer una neteja general, el logotip de la Universitat Autònoma i un botó per tancar sessió, on ambdós tenen el mateix propòsit. Per la mobilitat entre els diferents apartats del panell de control (Fig. 5), tindrem un menú desplegable a la part lateral (Fig. 13, Apèndix).

7 LLANÇAMENT A PRODUCCIÓ EN UN ENTORN REAL

Una vegada el codi ha acabat el desenvolupament inicial i passa tots els tests, aquest es va posar producció per a l'ús per part dels estudiants de Fonaments de Xarxes a les classes de laboratori.

En diverses sessions de 12 a 24 alumnes cadascuna organitzats en grups de fins a quatre integrants, els estudiants han creat els seus entorns de virtualització mitjançant la plataforma Web, i gràcies a la configuració enviada per correu del túnel VPN WireGuard, han aconseguit connectar-se amb èxit als seus entorns virtualitzats.

A la primera sessió, únicament dos dels grups no van aconseguir connectar-se a la seva màquina virtual. Tot i que la màquina virtual del grup es va generar correctament, a aquesta no es va assignar la seva VLAN del túnel VPN, i per tant no tenien connexió des de l'exterior. Després d'analitzar el problema, es va concloure que es tractava d'un error de threading a l'API de Proxmox.

Per tal de solucionar el problema, ara es realitza una segona comprovació de què la VLAN estigui assignada abans

d'encendre la màquina, i en cas contrari, s'assigna. Aquesta solució elimina per complet el problema.

Una vegada solucionat aquest problema, a la segona sessió de pràctiques, alguns grups no podien crear el seu entorn virtualitzat.

Es va identificar que només els últims grups tenien aquest problema, i es va arribar a la conclusió que s'havia arribat al nombre màxim de màquines virtuals.

El problema era que el nombre de VLAN assignades màxim és 100, però no es reaprofitaven les VLAN de les màquines eliminades. A l'haver arribat a aquest límit, no s'estaven assignant les VLAN fora del rang permès a les màquines virtuals i, per tant, no tenien accés al túnel.

En reaprofitar aquestes VLAN que ja no estaven assignades a cap màquina, aquest problema se soluciona i es dona per completat el desenvolupament del projecte, amb un funcionament correcte a la resta de sessions de pràctiques.

8 PRESENTACIÓ I DISCUSSIÓ DELS RESULTATS

Seguint la planificació, el desenvolupament ha acabat dins dels terminis de temps previstos. S'ha aconseguit elaborar la plataforma Web que hem vist a la implementació, el backend que gestiona la infraestructura de màquines virtuals i la interconnexió entre el frontend i el backend complint tots els requisits i objectius plantejats. Inclòs s'han arribat a implementar funcionalitats no plantejades a la planificació del projecte com el selector d'idiomes. Es considera que els resultats obtinguts són assolibles per a un enginyer informàtic, donat l'aprenentatge a la carrera sobre els dos llenguatges de programació utilitzats (Vue i Node.js).

Analitzant la planificació del projecte, un problema afrontat ha sigut l'overscoping, on una vegada acabades les funcionalitats essencials, s'han creat noves funcionalitats al Frontend com podria ser el selector d'idiomes o el mode clar / fosc. Per una banda, potser hauria estat una millor opció realitzar un testeig més exhaustiu abans de posar el codi en producció. Per altra banda, aquest fet ha permès entregar més funcionalitats que aporten un valor al producte final.

Finalment, com s'ha esmentat en detall a l'apartat anterior, el llançament a producció als laboratoris de Fonaments de Xarxes ha sigut tot un èxit, i una vegada solucionats els petits problemes trobats s'ha observat un correcte funcionament a la resta de sessions, on els alumnes han pogut utilitzar la plataforma web per realitzar les pràctiques.

El projecte consta d'un total de més de 3700 línies de codi entre el backend, l'API REST i la interfície Web. Tot el codi s'ha fet públic a GitHub [24] per a la seva lliure consulta, modificació i ús amb més de 250 commits.

9 CONCLUSIONS

TECTONIC és un projecte d'innovació docent a les assignatures de xarxes de telecomunicacions que planteja remodelar el sistema actual d'ensenyament als laboratoris. Aquesta remodelació planteja la creació d'una plataforma on els estudiants puguin crear i gestionar xarxes virtualitzades de complexitat d'una forma senzilla.

Donada la magnitud del sistema, al punt de partida es compta amb la infraestructura física ja implementada.

Aquesta inclou: els servidors, que ja han sigut connectats i configurats, les màquines virtuals, que ja tenen la seva plataforma d'administració i les imatges corresponents i les interconnexions i configuracions bàsiques de l'encaminador.

El treball de fi de grau ha consistit en elaborar Nethermir, una eina de gestió i automatització de servidors i màquines virtuals per ajudar en la docència de les assignatures de xarxes de telecomunicacions, en la que es permetès als estudiants crear i connectar-se mitjançant un túnel VPN a les seves màquines virtuals per a la creació d'entorns de xarxa virtualitzats.

Els objectius principals eren, crear un programari de gestió de servidors backend, una plataforma Web per a interactuar amb el backend, i una API que permetès la comunicació entre els dos sistemes.

S'ha aconseguit elaborar una plataforma Web mitjançant Vue, en la que els estudiants poden crear i gestionar les seves màquines virtuals, i el professorat pot gestionar l'estat global dels servidors, que inclou la base de dades i l'estat de les màquines virtuals. L'entorn backend s'ha desenvolupat en node.js, un entorn asíncron que permet les connexions simultànies de tots els estudiants sense cap problema.

Una vegada implementades totes les funcionalitats, s'ha posat en producció el projecte, provant el sistema en un entorn real amb els estudiants de Fonaments de Xarxes amb un resultat final satisfactori.

Com a treball futur, encara es podrien afegir moltes funcionalitats, com accés a les màquines virtuals per l'entorn Web, un major nombre d'eines de gestió per part del professorat, o permetre als estudiants la creació de més d'una única màquina virtual. Es podria modificar el sistema actual per a adaptar-lo a altres assignatures de la carrera o fins i tot d'altres graus.

En últim lloc, es considera l'elaboració del projecte ha acabat amb un resultat final satisfactori. El projecte té grans implicacions en les pràctiques de laboratori de les assignatures de xarxes de telecomunicacions i el desenvolupament futur del projecte no té límits, respecte a funcionalitats i a altres entorns universitaris interessats a implementar aquest programari en les seves pràctiques de laboratori.

REFERÈNCIES

- [1] "Nethermir - panell de control." [Online] <https://nethermir.uab.cat> Últim accés: 06/02/2024.
- [2] R. I. Dinita, G. Wilson, A. Winckles, M. Cirstea, and A. Jones, "A cloud-based virtual computing laboratory for teaching computer networks," 2012.
- [3] T. Holterbach, T. Bühler, T. Rellstab, and L. Vanbever, "An open platform to teach how the internet practically works," 2019.
- [4] "React - the library for web and native user interfaces." [Online] <https://www.react.dev> Últim accés: 06/02/2024.
- [5] "React - the library for web and native user interfaces." [Online] <https://www.angular.io> Últim accés: 06/02/2024.
- [6] "Vue.js - the progressive javascript framework." [Online] <https://www.vuejs.org> Últim accés: 06/02/2024.
- [7] "Vmware esxi - what is esxi, bare metal hypervisor." [Online] <https://www.vmware.com/products/esxi-and-esx> Últim accés: 06/02/2024.
- [8] "Xen - bring the power of virtualization everywhere." [Online] <https://www.xenproject.org> Últim accés: 06/02/2024.
- [9] "Proxmox - powerful open-source server solutions." [Online] <https://www.proxmox.com> Últim accés: 06/02/2024.
- [10] "Bull - redis-based queue for node." [Online] <https://www.npmjs.com/package/bull> Últim accés: 06/02/2024.
- [11] "Kue - priority job queue backed by redis." [Online] <https://www.npmjs.com/package/kue> Últim accés: 06/02/2024.
- [12] "Bee - a simple, fast, robust job/task queue for node.js." [Online] <https://www.npmjs.com/package/bee-queue> Últim accés: 06/02/2024.
- [13] "Benchmarks for bee performance compared to other queue managers." [Online] <https://github.com/bee-queue/bee-queue/blob/master/benchmark/results-2017-08-12/resultschart.png> Últim accés: 06/02/2024.
- [14] "Nodemailer." [Online] <https://www.nodemailer.com> Últim accés: 06/02/2024.
- [15] "Console - simple debugging console." [Online] <https://www.nodejs.org/api/console.html> Últim accés: 06/02/2024.
- [16] "Pino - super fast, all natural json logger." [Online] <https://www.npmjs.com/package/pino> Últim accés: 06/02/2024.
- [17] "Winston - a logger for just about everything." [Online] <https://www.npmjs.com/package/winston> Últim accés: 06/02/2024.
- [18] "Express - node.js web application framework." [Online] <https://expressjs.com> Últim accés: 06/02/2024.
- [19] "Mysql." [Online] <https://www.mysql.com> Últim accés: 06/02/2024.
- [20] "node proxmox - a node.js client for proxmox." [Online] <https://www.github.com/ttarvis/node-proxmox> Últim accés: 06/02/2024.
- [21] "node-routeros - mikrotik routerboard api for node.js." [Online] <https://www.npmjs.com/package/node-routeros> Últim accés: 06/02/2024.
- [22] "Redis - a vibrant, open source database." [Online] <https://www.redis.io> Últim accés: 06/02/2024.
- [23] "Wireguard - fast, secure vpn tunnel." [Online] <https://www.wireguard.com> Últim accés: 06/02/2024.
- [24] "Nethermir - github repository." [Online] https://github.com/IvanGomezGon/nethermir_wbpage Últim accés: 06/02/2024.

APÈNDIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
5.1 Anàlisi i disseny														
5.2 Interfície Web														
• 5.2.1 Registre i inici de sessió														
• 5.2.2 Gestor de galetes de sessió														
• 5.2.3 Panell control professorat														
• 5.2.4 Panell control estudiant														
5.3 API REST														
5.4 Gestor de servidors														
• 5.4.1 Base de dades														
• 5.4.2 Gestor de galetes de sessió														
• 5.4.3 Administrador de MV														
• 5.4.4 Autoconfigurador encaminador														
• 5.4.5 Servidor de correus														
• 5.4.6 Logger rotatiu personalitzat														

Fig. 6: Diagrama de Gantt

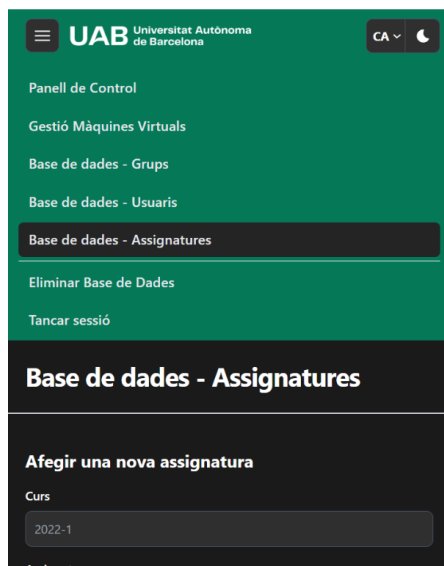


Fig. 7: Panell de control del professorat amb el menú superior (versió mòbil) obert.

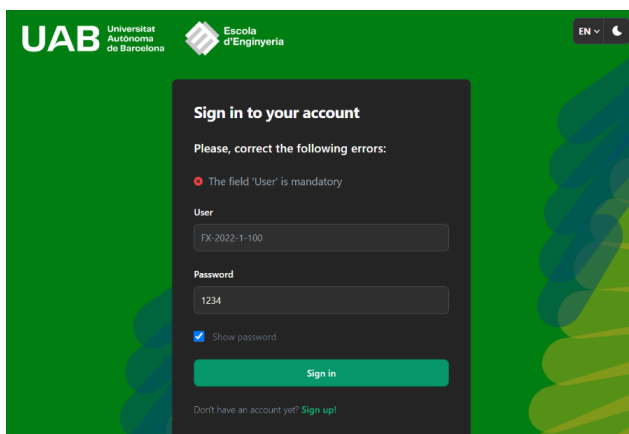


Fig. 8: Formulari d'inici de sessió, mode fosc. S'intenta iniciar sessió sense usuari.



Fig. 9: Pantalla que es mostra una vegada el grup inicia sessió per primer cop.

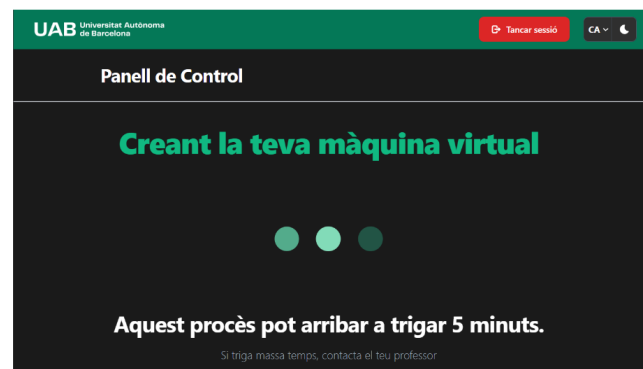


Fig. 10: Missatge d'espera mentre es clona la màquina.

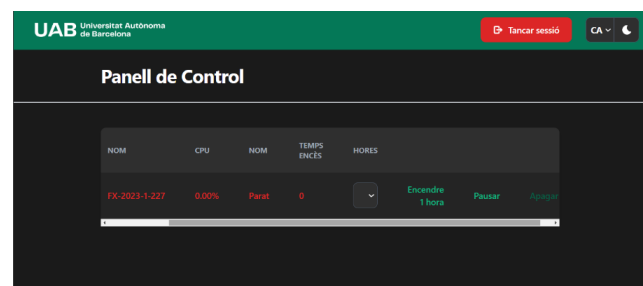


Fig. 11: Panell de control de l'estudiant una vegada s'ha acabat el procés de clonar.

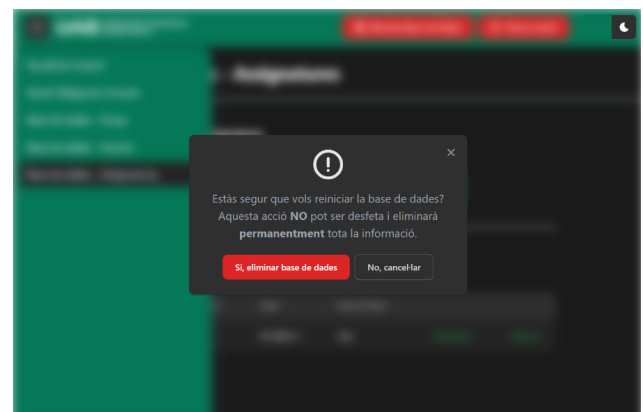


Fig. 12: Missatge mostrar en polsar a "Eliminar base de dades"

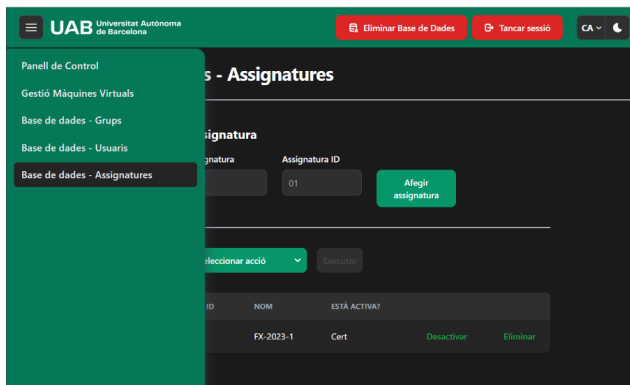


Fig. 13: Panell de control del professorat, amb el menú lateral obert.