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

Ramos Herrero, Christian; Robles Martínez, Sergi, dir. Desenvolupament d'un framework per al desplegament d'aplicacions web integrades en el SSO de la UAB. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/248499>

under the terms of the  license

Desenvolupament d'un framework per al desplegament d'aplicacions web integrades en el SSO de la UAB

Christian Ramos Herrero

Resum—Els programaris utilitzats per desenvolupar serveis web són molt diversos. Per aquest motiu, no tots els servidors són capaços d'allotjar tots els serveis web alhora, ja que depenent dels programaris emprats, no es poden aïllar tots els serveis web. Aquest fet limita que un servidor pugui ser compatible amb la utilització de diferents versions del mateix programari. En aquest treball de fi de grau es presenta un anàlisi sobre el tipus d'infraestructura que es pot utilitzar en un servidor d'un departament de la universitat que necessita córrer diferents serveis web desenvolupats per alumnes diferents. Aquest anàlisi acaba determinant que l'ús de la virtualització basada en contenidors és la millor opció respecte les altres per la millora en l'eficiència, rapidesa de muntatge, facilitat d'ús i possibilitat d'aïllament dels serveis. Com a resultat es presenta un projecte que determina el desenvolupament d'un framework per la configuració d'aplicacions web integrades en el SSO de la UAB.

Paraules clau— Apache, Docker, Framework, MariaDB, PHP, Single sign-on, Stack, Swarm.

Abstract—The set of programmes used in order to develop web services are many. For this reason, not all facilities are capable of hosting all the web services together since, depending on the programs used, all the services cannot be isolated. This fact limits the use of different software versions in accordance with the web service. This final degree project analyzes different structures that could be used in a university department's server that needs to run several web services developed from different alumni. This study ends up determining that the usage of the container's based virtualization is the best fitted option in comparison with the other alternatives for several reasons such as the increase in efficiency, the easy assembly, the usability and the possibility to isolate all the services. As a result, a project is presented concerning the development of a framework for the configuration of built-in web applications in the UAB's SSO.

Index Terms— Apache, Docker, Framework, MariaDB, PHP, Single sign-on, Stack, Swarm.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EL Departament d'Enginyeria de la Informació i de les Comunicacions (DEIC) de la Universitat Autònoma de Barcelona (UAB) disposa d'un servidor on hi estan allotjats diferents serveis web que van desenvolupat els alumnes al llarg dels anys en el seu treball de fi de grau (TFG) i on s'hi troben també els treballs que s'han posat en producció per proporcionar diferents serveis als alumnes i professors de la universitat. Aquest servidor permet, per tant, l'afegiment i eliminació de serveis web, l'exportació de treballs cap a altres servidors o màquines virtuals i l'accés als arxius per part dels alumnes des dels ordinadors dels laboratoris del departament o de forma remota.

Amb el pas dels anys, els diferents projectes que desenvolupen els estudiants s'han fet a partir de tecnologies d'última generació i de versions de programari més actualitzades, causant que aquest servidor s'hagi d'actualitzar a les últimes versions i tecnologies per tal d'adaptar-se a la demanda dels

estudiants. Aquest fet ha provocat que existeixi un problema d'incompatibilitat dels nous projectes amb els antics, ja que aquests últims no es poden actualitzar a les noves versions perquè podrien deixar de funcionar correctament.

Actualment el departament es troba immers en un procés de trasllat d'aquest servidor cap a un nou adquirit recentment i es vol aprofitar aquest moment per construir un sistema que solucioni la problemàtica en les versions dels programaris i que implementi l'autenticació d'usuari mitjançant el single sign-on (SSO) de la UAB, permetent als alumnes desenvolupar serveis més específics. Addicionalment es vol fer que aquest nou servidor permeti l'afegiment i eliminació ràpida d'aplicacions web, que proporcioni flexibilitat als alumnes per utilitzar diferents tecnologies i es volen mantenir les característiques de l'anterior servidor que permetien exportar treballs i accedir als fitxers des de qualsevol ordinador dels laboratoris del departament o remotament.

L'objectiu, per tant, és crear un servidor d'aplicacions web que sigui el màxim flexible possible en l'ús de tecnologies i versions de programari, que tingui una bona usabilitat i que compleixi els requisits que proposa el departament. En la figura 1 s'il·lustra de forma gràfica el diagrama físic del sistema, el qual defineix quin és

- E-mail de contacte: Christian.ramosh@e-campus.uab.cat
- Menció realitzada: *Tecnologies de la Informació*
- Treball tutoritzat per: *Sergi Robles Martínez (DEIC)*
- Curs 2020/21

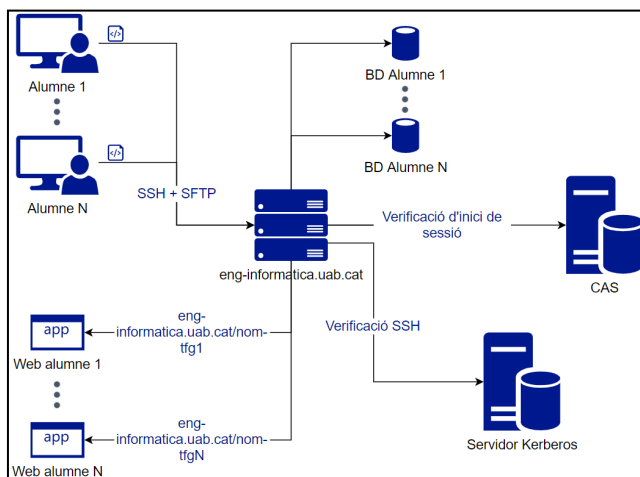


Fig. 1. Diagrama físic del sistema desitjat.

l'objectiu de muntatge d'aquest servidor. Els rols que el conformaran seran els desenvolupadors, els quals seran els alumnes que pujaran els fitxers web al servidor, els administradors, que seran qui gestionin les aplicacions web actives (normalment professors) i els usuaris, que seran qui visualitzaran els serveis web (alumnes i professors).

1.1 Objectius del projecte

A continuació es presenta l'objectiu que es vol complir amb la realització d'aquest projecte.

L'objectiu principal és desenvolupar un framework per la configuració d'aplicacions web integrades en el SSO de la UAB. Per arribar a aquest objectiu, s'han definit una llista de subobjectius prioritzats per ordre de preferència, els quals es presenten a continuació:

- Proveir un framework escalable i mantenible al llarg dels anys.
- Dissenyar, desenvolupar i desplegar piles de serveis web independents entre si.
- Dissenyar, desenvolupar i desplegar un sistema que possibiliti l'inici de sessió en el SSO de la UAB.
- Dissenyar, desenvolupar i desplegar eines que facilitin la gestió del framework (scripts i fitxers d'ajuda).

1.2 Escenari

A continuació es presenta l'escenari en el que es despleguen els serveis web actualment, així com una comparació de les diferents tecnologies que s'utilitzen tenint en compte el tipus de servidor que es vol desplegar en aquest projecte.

Pel que fa a l'escenari d'avui dia, per tal de desplegar serveis web normalment es requereix d'un sistema operatiu, un servidor web, un gestor de bases de dades i un llenguatge de programació. Totes aquestes tecnologies formen una stack de serveis web i requereixen d'un sistema que les faci funcionar. Aquest sistema, per tant, es pot muntar utilitzant màquines virtuals, directoris o contenidors.

Pel que fa a les màquines virtuals, tenen com a avantatges que poden funcionar per sobre de qualsevol sistema operatiu, proporcionen una fàcil exportació de les dades per posar en producció els serveis web i permeten que cada servei web utilitzi tecnologies i versions diferents. Com a inconvenients, aquesta tecnologia té un alt ús de recursos, el seu desplegament és lent, la configuració de les màquines és complexa i ocupen un alt espai d'emmagatzematge.

Pel que fa al sistema amb directoris, com a avantatges té que es tracta d'un sistema poc complex, que proporciona la capacitat de reaprofitar els recursos comuns i que permet utilitzar diferents tecnologies alhora. Com a inconvenients, hi ha la difícil gestió de la configuració en cas d'utilitzar diferents tecnologies i la complexa exportació de les dades dependent de la tecnologia.

Pel que fa al sistema amb contenidors, com a avantatges té que és un sistema que permet un fàcil desplegament dels serveis, permet la possibilitat d'utilitzar tecnologies i versions diferents per cada servei web, té una alta escalabilitat, facilita l'exportació de les dades, facilita la gestió de la configuració dels serveis i permet una gestió senzilla dels recursos a través de comandes. Com a inconvenient, aquesta tecnologia té un alt ús de recursos i emmagatzematge en cas que els diferents serveis web no puguin aprofitar els mateixos recursos i tecnologies.

Per tant, les alternatives més viables que existeixen per tal de complir amb els objectius proposats són el sistema de màquines virtuals i el sistema de contenidors.

1.3 Metodologia

A continuació s'exposa la metodologia utilitzada per la realització d'aquest projecte, la justificació de la utilització de la mateixa i el rendiment que ha donat durant les primeres i últimes fases del projecte.

La metodologia escollida ha estat Kanban perquè és un mètode lleuger, flexible, que no té rols definits i augmenta el rendiment general del projecte. Això es veu reflexat en que aquesta metodologia facilita i millora la productivitat del projecte, la classificació de les tasques, la visualització del flux de treball, la mesura del temps d'entrega i l'avaluació contínua d'oportunitats de millora. Els avantatges principals de l'ús d'aquesta tecnologia en aquest projecte vers d'altres és que s'adapta a la forma de funcionar d'un treball de fi de grau ja que sols està realitzat per una persona i a diferència d'altres metodologies com Scrum, en aquesta no es requereix una definició de rols. També proveeix la capacitat d'una millora continua ja que la realització d'aquest projecte s'ha de fer de forma iterativa, és a dir, a cada fase hi haurà com a resultat un prototip del que serà la versió final del producte. I per últim, aquesta metodologia proporciona una alta flexibilitat, degut a que obre la porta a canviar les prioritats de les tasques si el departament ho considera oportú degut a una necessitat urgent i permet ajustar-se millor al temps en el que s'ha de realitzar el projecte, ja que si per algun cas no dona temps a complir tots els

objectius d'aquest, com a mínim es tindrà una versió funcional del projecte.

Per tal de dur-la a terme s'ha utilitzat l'eina en línia de Trello [1], la qual proporciona una interfície gràfica en la que es pot construir una pissarra de Kanban i es poden moure els elements fàcilment per canviar-los d'estat (p.e de "To Do" a "Doing").

Pel que fa a l'ús de la metodologia durant les primeres fases del projecte, ha resultat ser molt senzilla d'utilitzar ja que ha ajudat a organitzar de forma visual totes les tasques que quedaven pendents i quines s'estaven realitzant en cada moment, així com saber en quina data s'havien de finalitzar. Per una altra banda, es tracta d'una metodologia molt flexible ja que ha permès iniciar certes tasques de forma paral·lela a altres, com és el cas de la generació de documentació (s'ha estat completant aquesta tasca mentre es realitzaven les altres) i ha permès avançar certes tasques degut a que d'altres s'han vist endarrerides per motius externs, com és el cas de la subtasca de configuració del servidor per permetre l'inici de sessió amb el SSO de la UAB.

Pel que fa a l'ús de la metodologia durant l'etapa final del projecte, s'ha consolidat la utilització de la metodologia Kanban ja que ha permès cada cop organitzar la informació de forma més senzilla i la utilització del Trello ha estat clau per fer-ho tot més senzill.

Per tant, l'ús d'aquesta metodologia ha resultat en una millor organització del treball i ha servit per avançar en les tasques de forma més eficient degut a que el projecte estava format per diferents fases i cada fase requeria centrar-se en treballar uns aspectes concrets i diferents de la resta de fases.

1.4 Planificació del treball

En aquest apartat es presenta la planificació establerta per la realització d'aquest projecte així com les modificacions que aquesta va sofrir degut a una adaptació a la forma de treball realment duta a terme.

Es van definir un total de 29 tasques classificades en 3 grups (Planificació, Elaboració del projecte i Tancament del projecte) i es van programar fins a 7 entregues. Pel que fa al grup d'Elaboració del projecte, es van definir 5 fases les quals corresponien als 5 prototips que es definirien en el projecte i que anirien millorant el producte final. Aquests incloïen un desenvolupament i una verificació del correcte funcionament d'aquest. Les fases definides no es solapaven entre elles excepte en el cas de la fase 3, la qual es feia de forma paral·lela a la resta de fases.

Pel que fa al calendari, la inicialització del projecte es va establir pel dia 15 de març i la seva finalització pel dia 12 de juliol, és a dir, la duració total del projecte ha estat de 17 setmanes. En l'apèndix es mostren el llistat de tasques d'aquest projecte i el diagrama de Gantt resultant de la planificació del treball.

Pel que fa a les modificacions fetes en la planificació, inicialment el projecte es va definir de forma diferent a la descrita ja que es va plantejar un model en cascada que pretenia anar avançant en les tasques progressivament i no tenir els resultats del projecte fins a la finalització del

mateix. Les conseqüències de l'avenç del projecte, però, van provocar que la planificació sofrís certs canvis quan el projecte ja estava en marxa per tal d'adaptar-se a la forma treball utilitzada finalment. Aquests canvis van consistir en modificar l'estructura anterior d'organització amb 4 fases definides per convertir el projecte en una organització per prototips (o fases) ja que s'ajustava millor a la forma en la que s'estava treballant realment en el projecte.

També cal destacar que es va decidir no realitzar la subtasca de "Realització d'un estudi ètic i legal" degut a que es va considerar innecessària en aquest projecte, ja que en l'informe de viabilitat tècnica es va determinar que la realització d'aquest no tenia implicacions ètiques directes en la societat, que tot el software utilitzat era de codi obert i que no caldria ajustar el projecte a cap mena de llei degut a que no es realitzava un tractament de dades privades i no es recollien dades a través de cookies o altres mètodes homòlegs.

Per últim, va haver-hi ajustos pel que fa a l'inici de les tasques. Això va ocórrer en el cas de la subtasca de "Configuració del servidor per permetre l'inici de sessió amb el SSO de la UAB". En aquesta subtasca s'havien configurat a temps els arxius necessaris per tal de fer funcionar el sistema d'autenticació centralitzat, però es va veure retrassada degut a que els serveis de SSO de la UAB encara no havien autoritzat de forma correcta al servidor del departament per tal de poder realitzar els inicis de sessió. Per tant, les comunicacions amb els responsables d'aquest servei van retrassar aquesta tasca fins al punt de començar a iniciar d'altres que no depenien de l'avenç d'aquesta.

Per tant, la planificació del projecte ha sofrit modificacions al llarg de l'avenç de les tasques, evidenciant una planificació inicial errònia i que no s'ajustava a la realitat d'un projecte d'aquesta índole.

2 ESTAT DE L'ART

En aquest apartat es presenta l'estat de l'art (en el moment de la realització d'aquest projecte) de les alternatives més viables per la realització del projecte (sistema amb màquines virtuals i sistema amb contenidors), presentant dos projectes de camp que s'han dut a terme i que han servit d'inspiració per aquest projecte.

Per una banda es recull el treball realitzat per Mathijs Jeroen Scheepers [2] que posa èmfasi en la comparativa de les tecnologies de màquines virtuals i contenidors. En aquest es proposa la virtualització basada en contenidors com una alternativa a la virtualització tradicional ja que aconsegueix reduir la sobrecàrrega dels centres de dades. Tot això ho justifica presentant els resultats d'una comparació de rendiment entre les dues implementacions, així com una discussió sobre la seva flexibilitat operativa, arribant a la conclusió que es poden aprofitar millor els recursos amb un sistema de contenidors si es volen executar molts processos aïllats entre ells.

Per una altra banda, el treball de Bret Fisher [3] presenta un exemple pràctic d'un projecte d'una aplicació de votació realitzat amb tecnologia Docker [4] (contenidors) on configura un sistema amb múltiples contenidors amb Docker compose [5] on tots es comuniquen entre si per transmetre's informació. Es tracta d'un projecte flexible ja que es pot desplegar a través de Docker, a través de Docker Swarm [6] o a través de Kubernetes [7]. El projecte està format per 5 contenidors, on hi ha un contenidor que proporciona una interfície front-end programada en Python, la qual permet votar a l'usuari entre 2 opcions, un contenidor amb Redis que recollia les noves votacions, un contenidor .NET Core que consumeix els vots i els emmagatzema en un altre contenidor amb una base de dades Postgres i un contenidor amb Node.js que mostra els resultats de la votació en temps real. Aquest muntatge és molt similar al que es podria implementar en algun treball d'un desenvolupador i per tant proporciona una visió molt àmplia sobre com es podria aprofitar la tecnologia Docker per suportar diferents serveis web, els quals estarien aïllats entre ells i suportats per varis dockers, permetent als desenvolupadors la construcció del seu servei web sense afectar als altres.

Com a conclusió, es pot extreure que ambdues tecnologies representen una bona opció ja que proporcionen portabilitat, aïllament i optimitzen la utilització dels recursos hardware però depenent del tipus de muntatge, serà preferible utilitzar una tecnologia o altra.

3 DESENVOLUPAMENT

3.1 Anàlisi de requisits funcionals

A continuació s'especifica el llistat de requisits funcionals que s'han recollit per tal de realitzar el projecte.

- El servidor ha de permetre la identificació d'usuaris mitjançant el SSO de la UAB.
- El framework ha de permetre als desenvolupadors accedir als seus serveis web a través d'enllaços i directoris.
- Els desenvolupadors han de poder accedir a la seva base de dades utilitzant un usuari i contrasenya.
- Els desenvolupadors han de poder accedir al seu directori de treball de forma remota utilitzant SSH i autenticant-se amb Kerberos [8].
- El framework ha de tenir la capacitat d'exportar a producció els serveis web dels desenvolupadors.

3.2 Anàlisi de requisits no funcionals

A continuació s'especifica el llistat de requisits no funcionals que s'han recollit per tal de realitzar el projecte.

- El servidor ha de fer ús de la virtualització.
- El framework s'ha de muntar sobre el sistema operatiu de Debian.

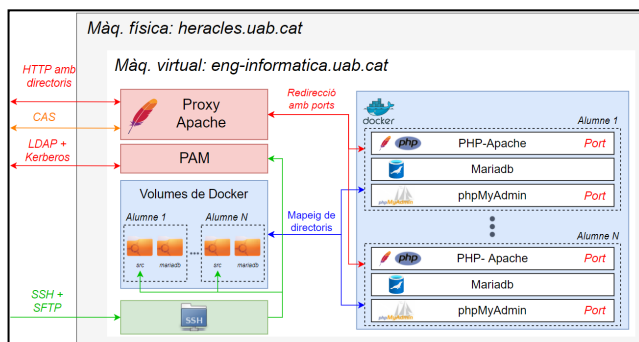


Fig. 2. Diagrama del servidor.

- El framework ha de suportar diferents versions del mateix programari.
- Cada desenvolupador ha de tenir un espai de desenvolupament aïllat dels altres per produir els seus serveis web.
- L'entorn de desenvolupament i visualització del web de cada desenvolupador ha d'estar sempre disponible.
- El servidor ha d'oferir serveis web en desenvolupament i producció.

3.3 Disseny

En aquest apartat es presenta el disseny finalment escollit un cop valorades les possibles alternatives i per quin motiu és millor que les altres opcions. També, dins d'aquest sistema es detallen els escenaris de desenvolupament i producció de serveis web i s'expliquen les dues possibles vies de construcció que existeixen del servidor.

La infraestructura del servidor incorporarà el programari de Docker degut a que és la tecnologia que permet utilitzar contenidors, que millor soluciona les problemàtiques trobades, que proporciona usabilitat i que facilita la incorporació de les noves funcionalitats desitjades pel departament. Tal i com s'ha vist anteriorment, aquesta tecnologia permet el muntatge d'un servidor on hi poden estar allotjats diferents serveis webs, tots aïllats entre ells. Això introdueix la possibilitat de muntar serveis web que utilitzin versions diferents del mateix programari i obre la porta a que els desenvolupadors puguin escollir quines tecnologies volen utilitzar pels seus serveis web sense afectar a la resta. A més a més, Docker incorpora també l'orquestrador Swarm, el qual permet posar programari en producció, facilitant la gestió si es despleguen diferents serveis, proporcionant escalat en cas que els serveis hagin de donar cobertura a molts desenvolupadors i proporcionant monitoratge en cas que hi hagi problemes, assegurant sempre la disponibilitat dels serveis. Aquesta tecnologia es combinarà amb un servidor Apache que distribuirà les peticions web dels usuaris cap al Docker que correspongui.

Totes aquestes funcionalitats també es podien aconseguir a través d'un sistema de màquines virtuals però la seva implementació provocaria, per una banda, que molts recursos del servidor (on estarien allotjats els

serveis web) es malbaratessin degut a que cada màquina virtual tindria assignats uns recursos per defecte i no s'estarien aprofitant tots i, per una altra banda, que el desplegament dels serveis fos molt costós en temps i emmagatzematge degut a la quantitat de recursos que s'han d'utilitzar per obrir una màquina virtual. És per aquest motiu que s'ha decidit escollir la tecnologia Docker.

Per últim, cal destacar que dins de la pròpia tecnologia de Docker existeixen dues formes de fer el muntatge del servidor. Una forma és utilitzar un contenidor per desenvolupador i l'altra és l'ús de l'eina Docker Compose, la qual utilitza múltiples contenidors per un mateix desenvolupador. Ambdues formes permeten muntar un servidor amb una stack de serveis web per cada desenvolupador però cada alternativa proporciona uns avantatges i inconvenients que s'han estudiat i valorat, havent arribat a la conclusió que l'eina més adient és Docker Compose. L'alternativa del muntatge d'un contenidor per desenvolupador facilita la gestió i identificació dels contenidors i redueix la quantitat d'emmagatzematge i recursos que s'utilitzen però a diferència de Docker Compose, la gestió de la configuració de cada contenidor que es generi és molt més difícil ja que la instal·lació dels serveis que tindrà el desenvolupador a disposició depèn d'un fitxer de configuració (Dockerfile) i dels paràmetres que es passin a la línia de comandes, motiu pel qual la fa una alternativa difícil de mantenir. En canvi, en Docker Compose cal tenir un fitxer de tipus YAML on s'especifiquen els serveis que correran per desenvolupador i un Dockerfile on s'hi especifiquen comandes addicionals que es vulguin executar abans d'inicialitzar la infraestructura per tal d'instal·lar, per exemple, llibreries o dependències que puguin ser necessàries. Una altra avantatge que proporciona Docker Compose és que si un programari té un error i el seu servei cau, la resta de serveis continuaran funcionant i no caldrà aturar tots els serveis per solucionar el problema, tal i com sí passaria en el cas de tenir tots els serveis dins d'un sol contenidor.

Per tant i com a conclusió, s'ha utilitzat la tecnologia Docker conjuntament amb l'eina Docker Compose per proporcionar als desenvolupadors que accedeixin al servidor un espai de desenvolupament aïllat i s'ha utilitzat la tecnologia de Docker Swarm per tal de desplegar el sistema en el servidor (s'utilitzen stacks de Docker Swarm). A la figura 2 es presenta el framework plantejat, el qual està format per grups de contenidors Docker, on cada un pertany a un desenvolupador diferent. Cada grup de contenidors té mapejat els fitxers de desenvolupament corresponents al contenidor PHP-Apache i les dades de MariaDB en un directori dins la màquina virtual i els desenvolupadors només tenen accés a la carpeta de desenvolupament via SSH. El sistema també té un servidor Apache que redirigeix les peticions HTTP (amb directoris) cap al contenidor corresponent utilitzant els ports on cada contenidor corre els seus

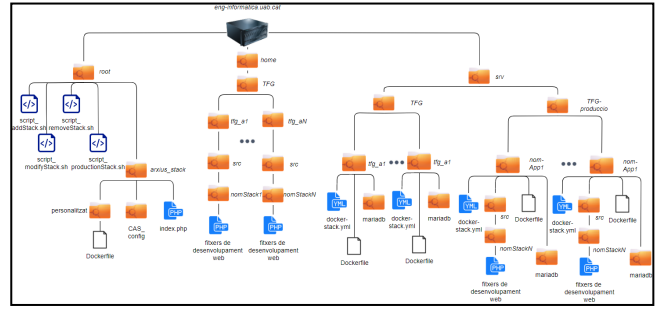


Fig. 3. Estructura de directoris del servidor. A /root hi ha els arxius de gestió, a /home els arxius de desenvolupament i a /srv els arxius de còpia de seguretat i de producció

serveis. Per últim, el servidor es connecta amb el servidor Kerberos del DEIC per poder verificar els desenvolupadors que es connectin via SSH.

Aquest servidor té com a configuració bàsica la stack de serveis web Linux, Apache, MariaDB i PHP (LAMP) en la seva última versió de programari i un fitxer d'exemple on es permeti fer un inici de sessió al SSO i es permet configurar una stack de serveis personalitzada en cas que el desenvolupador ho sol·liciti o el departament ho requereixi.

3.4 Exposició dels resultats

En aquest apartat s'expliquen i s'interpreten els resultats obtinguts del desplegament del projecte.

Un cop finalitzat el projecte, el resultat d'aquest és un servidor connectat a la xarxa el qual és capaç d'allotjar i proveir diferents serveis web a través del domini "eng-informatica.uab.cat". Això s'aconsegueix mitjançant una infraestructura amb un servidor Apache com a Proxy i amb contenidors Linux (Docker) on cada servei web disposa d'una stack de Linux, Apache, MariaDB i PHP configurades en 3 contenidors. Cada stack disposa d'aquests 3 contenidors, els quals són: PHP-Apache, MariaDB i PHPMyadmin, la versió dels quals i els paquets addicionals que es vol que tingui el PHP-Apache es poden escollir segons les preferències del desenvolupador. Per una altra banda, existeix també la possibilitat de configurar serveis web que utilitzin una stack diferent a la que el framework ofereix per defecte.

Pel que fa a la configuració de stacks i desenvolupament web, hi ha configurada la possibilitat de connectar-se remotament via SSH per tal de gestionar la configuració de stacks (pels administradors) i per penjar-hi els fitxers de desenvolupament (pels desenvolupadors). Un cop dins del servidor, aquest proveeix 4 scripts als administradors que permeten configurar, modificar, posar en producció o esborrar una stack de serveis web i proveeix un directori al desenvolupador per posar-hi els seus fitxers web. Per últim, existeix també una wiki de documentació pels administradors on s'hi troben 5 guies que detallen la instal·lació i configuració inicial del servidor, la creació d'una stack d'un desenvolupador des de zero, informació addicional sobre tots els programaris utilitzats per fer funcionar la infraestructura, com està fet el muntatge de la infraestructura i com ha de procedir el desen-

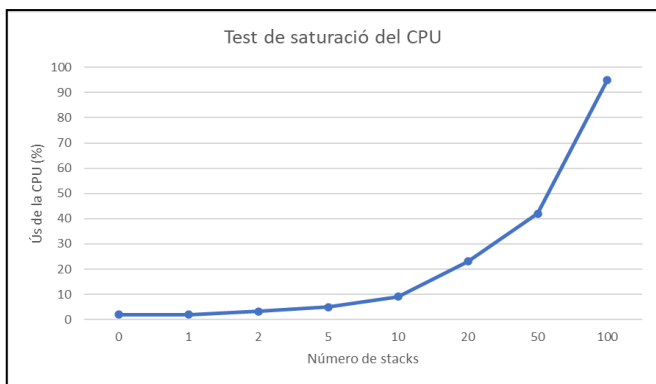


Fig. 4: Percentatge de saturació de la CPU segons es generen les stacks.

volupador per utilitzar el servidor.

Pel que fa a la infraestructura de directoris, per una banda hi ha el directori `"/home/TFG"` del servidor que s'organitza de forma que cada stack té un directori independent (tfg-XX) on s'hi troben els arxius relacionats amb el desenvolupament web (subdirector `"/src"`). El desenvolupador té accés al seu directori independent i pot modificar els arxius que es troben a dins. Aquesta carpeta està mapejada amb el contenidor PHP-Apache, que és el que interpreta els fitxers i els mostra al web. Per una altra banda, el servidor té uns altres dos directoris (`"/root"` i `"/srv"`) on, per una banda, a `"/root"` es troben els fitxers relacionats amb la gestió dels dockers (els scripts mencionats anteriorment i els arxius inicials necessaris) i per una altra banda, a `"/srv"` hi ha dos directoris més. El primer dels directoris és una còpia de seguretat (`"/TFG"`) dels fitxers utilitzats per crear la stack (docker-stack.yml i Dockerfile) i del directori `"/mariadb"` (on s'emmagatzemen les dades de la base de dades) organitzat de la mateixa forma que el directori `"/home/TFG"`. L'altre directori és a on s'hi allotjen els fitxers de les stacks que estan en producció (`"/TFG-producció"`), amb tots els fitxers i directoris necessaris junts (`"/src"`, `"/mariadb"`, docker-stack.yml i Dockerfile). A la figura 3 es mostra la forma en la que s'organitzen els 3 directoris mencionats en el servidor.

Aquesta infraestructura utilitza programari disponible en línia i de codi obert. Els desenvolupadors d'aquests programaris posen a disposició de qualsevol persona un conjunt de guies i recursos que permeten configurar-los segons cada necessitat. Per tant, els resultats obtinguts en aquest projecte són assolibles per a un enginyer informàtic tenint en compte els recursos disponibles.

3.5 Proves de rendiment del servidor

En aquest apartat es presenta de forma sintètica les proves en un entorn real que s'han realitzat en el servidor eng-informatica.uab.cat per tal de posar al límit els recursos del servidor. Totes les proves s'han realitzat mitjançant la creació de les stacks bàsiques del framework (conjunt de 3 contenidors amb la stack de programes LAMP: Linux, Apache, MariaDB i PHP).

Respecte el context de les proves realitzades, s'han analitzat els components hardware als quals té accés la

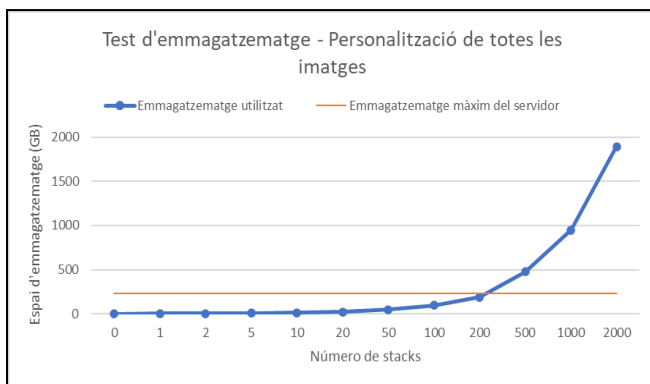


Fig. 5: Espai d'emmagatzematge ocupat segons es generen les stacks (canviant cada cop les 3 imatges).

màquina virtual que s'ha configurat en el servidor especialment per aquest projecte i s'ha comprovat el comportament de diverses variables segons s'augmenten el nombre de stacks simultànies en funcionament. Aquestes variables són l'esgotament de l'espai d'emmagatzematge, el nivell de saturació del processador i el temps de resposta del servidor a l'exterior. Totes les proves realitzades s'han fet a través d'un client SSH connectat al servidor eng-informatica.uab.cat en el port 169 i autenticat mitjançant certificats digitals. En totes elles s'han utilitzat scripts programats específicament per aquest objectiu que autogeneren tantes stacks com se li indiquin. Les dues primeres proves (emmagatzematge i saturació) provenen d'un únic test que s'ha realitzat sobre el servidor. Pel que fa a la última prova (temps de resposta), s'ha realitzat en 3 entorns diferents ja que es volien descartar que algunes dades haguessin pogut veure's alternades com a conseqüència d'algun fet extern aliè al servidor: el primer entorn era dins la pròpia xarxa del DEIC amb una velocitat de 1Gb/s, una altra s'ha realitzat en un entorn a partir d'una connexió de fibra simètrica amb una velocitat de 600Mb/s i per últim, s'ha realitzat en un entorn a partir d'una connexió de 4G amb una velocitat no simètrica i amb una velocitat de 40Mb/s de mitjana.

Pel que fa a la prova d'emmagatzematge i tenint en compte les 3 casuístiques analitzades (reaprofitament d'imatges, personalització de la imatge de PHP-Apache i personalització de les 3 imatges), s'extreu que en les dues primeres, l'emmagatzematge s'esgotaria entre les 1000 i les 2000 stacks i respecte la última prova s'extreu que l'emmagatzematge s'esgotaria entre les 200 i 300 stacks, reduint-se en 5x el moment en el que s'assoleixen el màxim de stacks màximes operant en el servidor.

Pel que fa a la prova de saturació del processador es conclou que l'augment en la saturació és contant segons s'afegeixen més stacks amb una mitjana de 2.02x d'augment, però al arribar a 100 stacks, el CPU deixa de generar contenidors i l'ocupació de processos d'aquesta és propera al 100%.

Pel que fa a la prova de temps de resposta del servidor s'ha constatat que el temps mitjà de les peticions varia en menys de 5ms segons s'augmenta el nombre de contenidors (la mitjana de temps és 41,71ms en els 3 entorns esmentats) però que quan s'arriba a 100 stacks aquest decau a 0 milisegons degut a que no respon a cap petició.

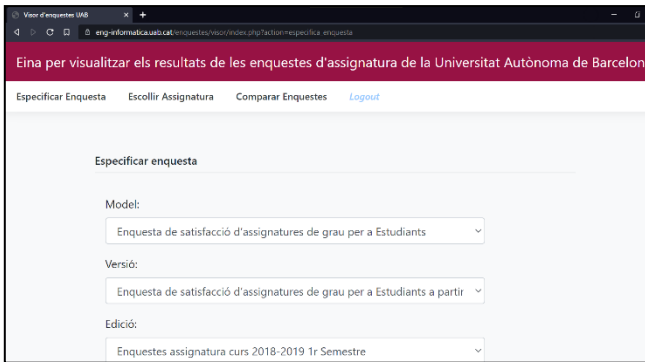


Fig. 6: Servei web “enquestes” desplegat en la prova pilot del framework. L'alumne ha desenvolupat un visor de resultats d'enquestes d'assignatura que requereix iniciar sessió en el SSO de la UAB.

Per tant i com a conclusió final pel que fa a les proves de rendiment, es pot determinar que el coll d'ampolla que impedeix que es puguin generar més de 100 stacks és el processador ja que aquest arriba al seu límit d'ús en aquesta xifra, tal i com es mostra a la figura 4. Com a solució a aquesta problemàtica en cas que es volgués augmentar la capacitat dels alumnes a més de 100, seria augmentar el número de nuclis als quals té accés la màquina virtual en el servidor per fer que aquest sigui capaç de suportar molts més contenidors en marxa, ja que al ser unitats independents, requereixen de múltiples processadors per funcionar. Per una altra, l'emmagatzematge seria el proper coll d'ampolla un cop solucionada la problemàtica anterior en el cas analitzat en que totes les imatges de cada stack fossin diferents ja que l'esgotament en la memòria es donaria entre les 200 i 300 stacks, tal i com es mostra en la figura 5. En aquest cas, per tant, la solució si es volguessin tenir més de 200 desenvolupadors simultanis en el servidor seria l'augment en l'espai d'emmagatzematge al qual té accés la màquina virtual en el servidor.

3.6 Prova pilot del framework

En aquest apartat es presenta de forma sintètica una prova pilot realitzada amb 2 desenvolupadors reals, alumnes d'un professor del departament DEIC, als quals se'ls ha entregat les seves credencials de Kerberos i una guia explicativa sobre com utilitzar el servidor. La prova, per tant, ha consistit en monitoritzar durant un temps de 2 setmanes la infraestructura i extreure una valoració de l'ús d'aquesta per part dels desenvolupadors.

Pel que fa a les seves aplicacions, s'han anomenat “enquestes” i “SPE” i aquestes han requerit d'una stack de serveis web bàsica amb Linux, Apache, MariaDB i PHP. Per tant, en primera instància, se'ls ha generat una stack a cadascun amb la última versió de cada imatge en el moment de fer la prova pilot. Aquestes versions han estat la 8.0.7 per PHP-Apache, la 10.5 per MariaDB i la 5.1.0 per PHPMyadmin. Acte seguit, se'ls ha entregat una guia en format PDF on se'ls explica el funcionament bàsic de la infraestructura i unes credencials per poder accedir mitjançant SSH al servidor `eng-informatica.uab.cat`, on han pogut emmagatzemar els seus fitxers web.

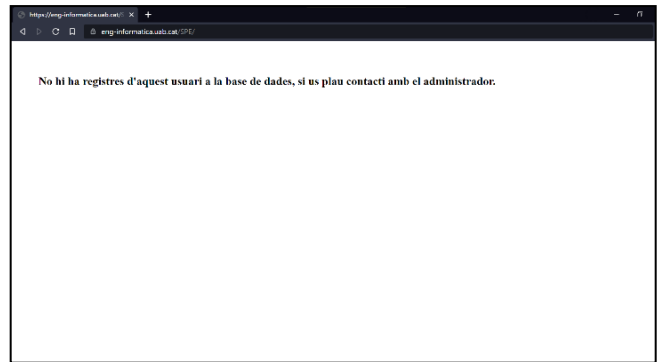


Fig. 7: Servei web “SPE” desplegat en la prova pilot del framework. L'alumne ha desenvolupat una eina de seguiment i tutorització de les pràctiques externes que filtra la informació segons l'usuari que ha iniciat sessió en el SSO.

Pel que fa als contenidors que mantenen les aplicacions web dels desenvolupadors, han utilitzat els ports 8082, 8083, 8084 i 8085 per les imatges de PHP-Apache i PHPMyadmin respectivament (els dos primers per una stack i els últims per l'altra). L'enllaç a on els desenvolupadors han pogut accedir als seus serveis web ha estat a través directoris, utilitzant el nom de la seva aplicació tal i com s'ha definit inicialment. Per tant, els seus enllaços han estat `https://eng-informatica.uab.cat/enquestes` i `https://eng-informatica.uab.cat/SPE`. Per una altra banda, els enllaços per accedir a la gestió de la base de dades (PHPMyAdmin) han estat a través del mateixos enllaços anteriors però afegint-hi la sentència “-phpmyadmin” al final. Per accedir a la base de dades, han necessitat unes credencials, que són l'usuari i contrasenya que se'ls ha proporcionat a l'inici. Un cop accedien a la gestió de la base de dades, han disposat d'un usuari que tenia permisos d'edició i lectura il·limitats sobre una base de dades que tenia el mateix nom que l'usuari.

Respecte la monitorització i la valoració de l'ús de la infraestructura, un dels desenvolupadors ha demanat la instal·lació de dues extensions de PHP (“gd” i “zip”) per necessitats del seu treball i se li han configurat utilitzant el script “script_modifyStack.sh”. Per una altra banda i per tal d'analitzar la usabilitat de l'eina i reportar possibles errors que hagin pogut sorgir, s'ha proporcionat una enquesta a cada desenvolupador. D'aquesta s'extreu que pel que fa al servidor, no s'ha reportat cap problema relacionat amb el rendiment ni caigudes i pel que fa a la usabilitat de l'eina, tots dos coincideixen que és molt senzilla i que la usabilitat és bona. Per últim, els dos desenvolupadors han esmentat que no tenen pensat demanar un canvi en els seus serveis, però que depenent de com evolucionin la seva aplicació web, ho podrien acabar necessitant, tal i com ha passat amb un dels dos.

Com a conclusió es pot extreure que la satisfacció d'ambdós desenvolupadors és prou bona i que el desenvolupament de les seves aplicacions web ha estat àgil i ràpid gràcies a poder-se connectar via SSH i utilitzar el programari PHPMyadmin per tal de gestionar els arxius i les dades de la base de dades. A les figures 6 i 7 es mostra el desplegament dels serveis web dels dos desenvolupadors, els quals autèntiquen l'usuari i discriminen els con-

tinguts que poden veure.

6 LÍNIES FUTURES DE DESENVOLUPAMENT

En aquest apartat es presenten les línies futures de desenvolupament un cop presentat el projecte actual i vistos els seus resultats en una prova real.

Aquest projecte ha nascut de necessitat del departament DEIC de tenir uns servidors que suportessin aplicacions web en desenvolupament i producció simultàniament. Per tant, l'objectiu principal del projecte ha consistit en pal·liar les necessitats més immediates de la construcció de la infraestructura i per tant s'ha plantejat un projecte enfocat a fer recerca de diferents alternatives per complir amb les necessitats més bàsiques i atendre els requisits mínims, deixant poc marge de maniobra per desenvolupar funcionalitats que milloressin la experiència d'usuari i la gestió de la infraestructura però deixant la porta oberta a desenvolupar tecnologies molt escalables.

Per una banda, els elements principals que manquen en aquest projecte i que poden servir com a línies futures de desenvolupament i millora del projecte són el disseny d'un middleware que permeti una gestió interactiva dels contenidors sense haver de connectar-se via SSH amb el servidor i executar els scripts donats. Aquest programari podria ser una pàgina web interactiva que donaria pas a que els professors que desitgin que els seus alumnes desenvolupin aplicacions web en el servidor `enginformatica.uab.cat` gestionessin de forma directa i senzilla els contenidors i les stacks, prescindint en tot moment de l'ajuda de qualsevol tècnic del departament. En aquesta línia, s'aconseguiria una millora en l'eficiència i inclús es podria donar accés al desenvolupador perquè ell mateix pogués fer els canvis que desitgés sense que això suposés un perill per la seguretat del servidor.

Per una altra banda, una altra línia futura de desenvolupament podria ser l'aprofitament de la tecnologia Swarm per orquestrar una framework distribuït en diferents servidors en la que es garantiria una alta disponibilitat dels serveis i en la que podrien participar-hi molts més desenvolupadors. Això seria possible degut a que Docker Swarm permet establir un node màster i nodes niats, els quals treballen tots junts per mantenir el sistema obert i disponible al llarg del temps, ja que si un s'apagués de sobte, Swarm s'encarregaria de garantir la disponibilitat dels serveis web en tot moment utilitzant la resta de nodes disponibles.

Per tant, les línies futures de desenvolupament passen per millorar la interfície d'usuari per fer la gestió de la framework més àgil, ràpida i eficient de cara als professors i als seus alumnes (els desenvolupadors) i per escalar els serveis que ofereix aquest framework a les grans masses generant, per exemple, un programari de codi obert que pogués allotjar treballs web d'alumnes de universitats d'arreu del món.

7 CONCLUSIONS

En aquest apartat s'exposen les conclusions que s'extreuen un cop desenvolupat el projecte.

Per una banda, pel que fa a la planificació del projecte, tota la investigació que s'ha dut a terme per tal de determinar quina seria la millor infraestructura que respongués a les necessitats ha estat un element clau, ja que ha permès posteriorment definir de forma molt ràpida com seria el framework un cop s'han escoltat les peticions i opinions del departament.

Per una altra banda, pel que fa al desenvolupament, aquest ha estat subjecte en tot moment a la opinió del departament, el qual sempre ha buscat la major disponibilitat i fàcil manteniment. Aquest fet ha provocat que s'escollís l'opció d'infraestructura amb contenidors i no en màquines virtuals i que hi haguessin canvis en els programaris previstos, com passar d'utilitzar docker-compose a utilitzar docker swarm.

Per tant, i com a conclusió, l'objectiu principal del projecte i tots els subobjectius derivats d'aquest s'han completat amb escreix degut a que els resultats obtinguts són els esperats i responen a la necessitat que tenia el DEIC en la configuració d'aquest nou servidor. Tot això s'ha fet complint sempre amb tots els requisits funcionals i no funcionals. En aquesta línia, totes les tasques s'han realitzat dins del període establert en la planificació i s'ha seguit la metodologia prevista, la qual ha permès un correcte desenvolupament de la infraestructura i una ràpida adaptació segons les problemàtiques que han sorgit.

Tot això ha fet que la realització d'aquest projecte hagi estat un èxit i que no hagi comptat amb cap element no tractat, objectiu no assolit i cap extensió del treball realitzat.

AGRAÏMENTS

En primer lloc, agraeixo al Departament d'Enginyeria de la Informació i de les Comunicacions i en especial a en Sergi Robles, Adrià Sánchez i Jordi Pons per haver-me donat la oportunitat de realitzar aquest treball de fi de grau en col·laboració amb el departament i també per fer propostes constructives del meu treball que m'han ajudat molt a aprendre com realitzar un projecte d'aquesta índole. Per una altra banda, m'agradaria agrair a la meua família i a la meua parella tot el suport que m'han brindat al llarg d'aquests 4 anys en el grau. Per últim m'agradaria dedicar aquest treball a tots els professors que ensenyen amb passió i vocació, deixant una empremta allà on passen i que han estat una font d'inspiració al llarg de la meua vida estudiantil, en especial a aquells que ja no estan. Aquest treball us el dedico a vosaltres, Lluís i Magda.

BIBLIOGRAFIA

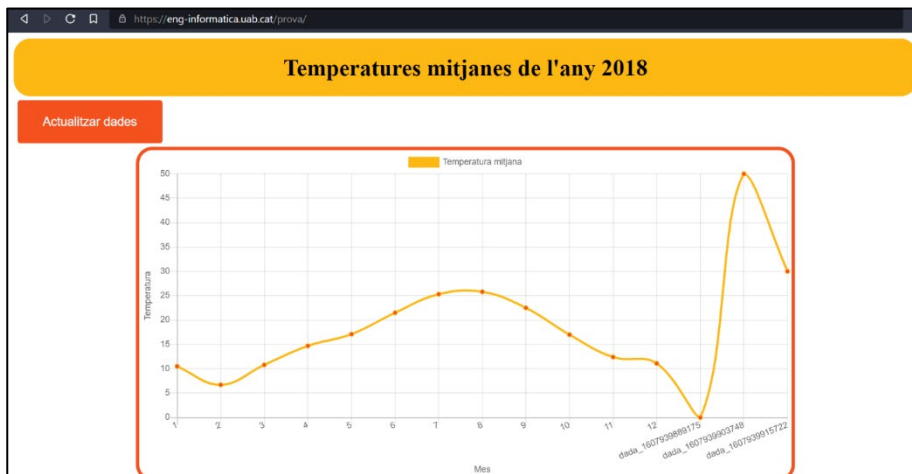
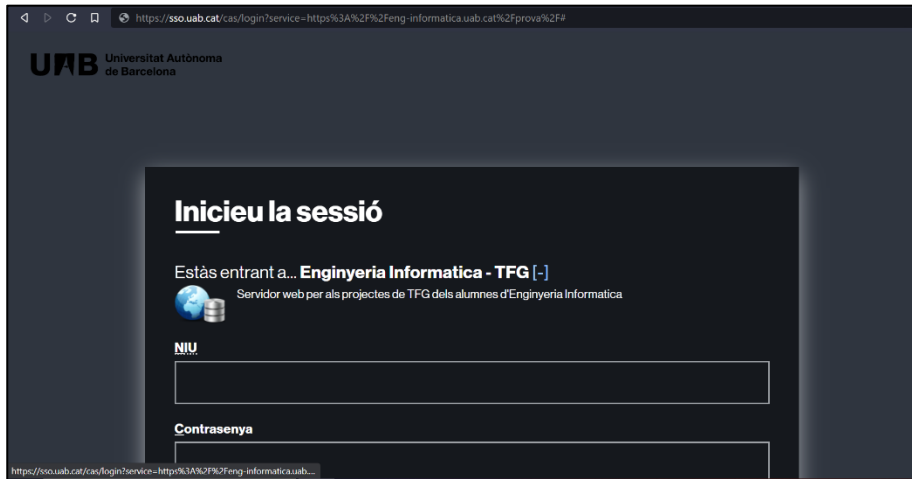
- [1] J. Spolsky, «Trello,» Atlassian, 2021. [En línia]. Available: <https://trello.com/>. [Últim accés: 5 4 2021].
- [2] M. J. Scheepers, «Virtualization and Containerization of Application,» de *21st twente student conference on IT*, vol. 21, 2014.

- [3] B. Fisher, «Example voting app,» 2016. [En línia]. Available: <https://github.com/BretFisher/example-voting-app>. [Últim accés: 28 2 2021].
- [4] S. Hykes, «Docker,» Docker, Inc., 13 Març 2013. [En línia]. Available: <https://www.docker.com/>. [Últim accés: 17 Abril 2021].
- [5] S. Hykes, «Docker compose,» Docker, Inc., 13 Març 2013. [En línia]. Available: <https://docs.docker.com/compose/>. [Últim accés: 17 Abril 2021].
- [6] S. Hykes, «Docker Swarm,» Docker, Inc., 13 Març 2013. [En línia]. Available: <https://docs.docker.com/engine/swarm/>. [Últim accés: 17 Abril 2021].
- [7] Google, «Kubernetes,» Google, 7 Juny 2014. [En línia]. Available: <https://kubernetes.io/>. [Últim accés: 17 Abril 2021].
- [8] S. P. Miller and B. C. Neuman and J. I. Schiller and J. H. Saltzer, «Kerberos authentication and authorization system,» de *Project Athena Technical Plan*, 1988.

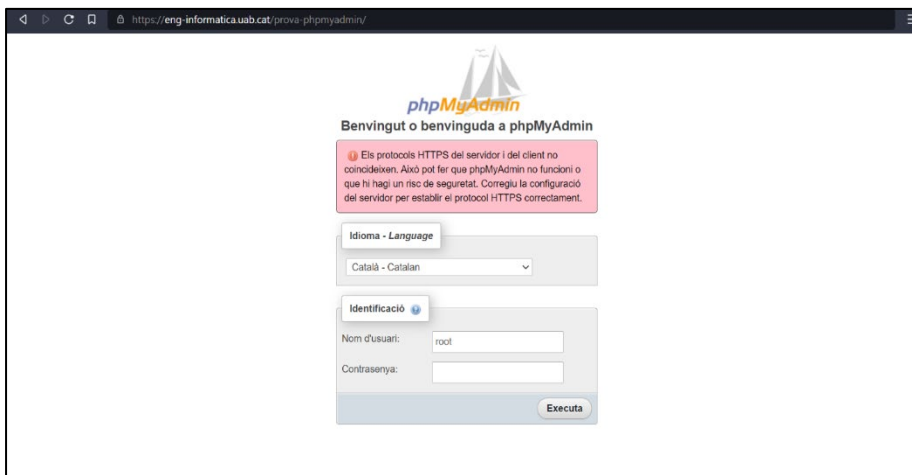
APÈNDIX

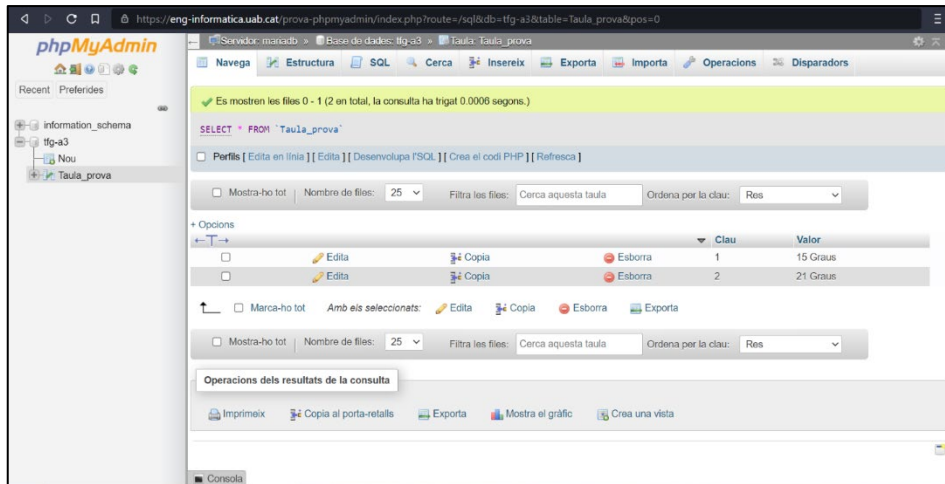
A1. Contingut web que proporciona el framework

Accés al contingut del contenidor PHP-Apache mitjançant el enllaç <https://eng-informatica.uab.cat/prova>



Accés al contingut del contenidor MariaDB mitjançant el contenidor de PHPMyAdmin a través de l' enllaç <https://eng-informatica.uab.cat/prova-phpmyadmin>





A2. Prova pilot de la infraestructura

Stacks generades pels desenvolupadors (tfg-b1 i tfg-b2) i de prova.

```

root@eng-informatica:~# docker stack ls
NAME      SERVICES ORCHESTRATOR
prova     3         Swarm
tfg-b1    3         Swarm
tfg-b2    3         Swarm
root@eng-informatica:~# docker stack services prova
ID        NAME                MODE             REPLICAS  IMAGE              PORTS
kd91klh8po8p  prova_mariadb      replicated        1/1        mariadb:10.5
51cp4qisf3wd  prova_php-apache   replicated        1/1        php-deic:8.0.6    *:8080->80/tcp
zhxltgnkx5bl  prova_phpmyadmin   replicated        1/1        phpmyadmin:5.1.0 *:8081->80/tcp
root@eng-informatica:~# docker stack services tfg-b1
ID        NAME                MODE             REPLICAS  IMAGE              PORTS
6psf0s10euar  tfg-b1_mariadb     replicated        1/1        mariadb:10.5
eo1f5s2x3ty0  tfg-b1_php-apache  replicated        1/1        php-deic:8.0.7    *:8082->80/tcp
cvmlr6lzkfnp  tfg-b1_phpmyadmin  replicated        1/1        phpmyadmin:5.1.1 *:8083->80/tcp
root@eng-informatica:~# docker stack services tfg-b2
ID        NAME                MODE             REPLICAS  IMAGE              PORTS
jfuupwb3dyv5  tfg-b2_mariadb     replicated        1/1        mariadb:10.5
n98odr0avolg  tfg-b2_php-apache  replicated        1/1        php-deic:8.0.7    *:8084->80/tcp
yaxqffpy500o  tfg-b2_phpmyadmin  replicated        1/1        phpmyadmin:5.1.1 *:8085->80/tcp

```

Distribució dels arxius en els directoris. A /home/TFG hi ha els fitxers web de les stacks en desenvolupament (tfg-b1 i tfg-b2), a /srv/TFG les dades de creació i de la base de dades de les stacks en desenvolupament (tfg-b1 i tfg-b2) i a /srv/TFG-produccio hi ha els fitxers web i les dades de creació i de la base de dades de les stacks de producció (prova).

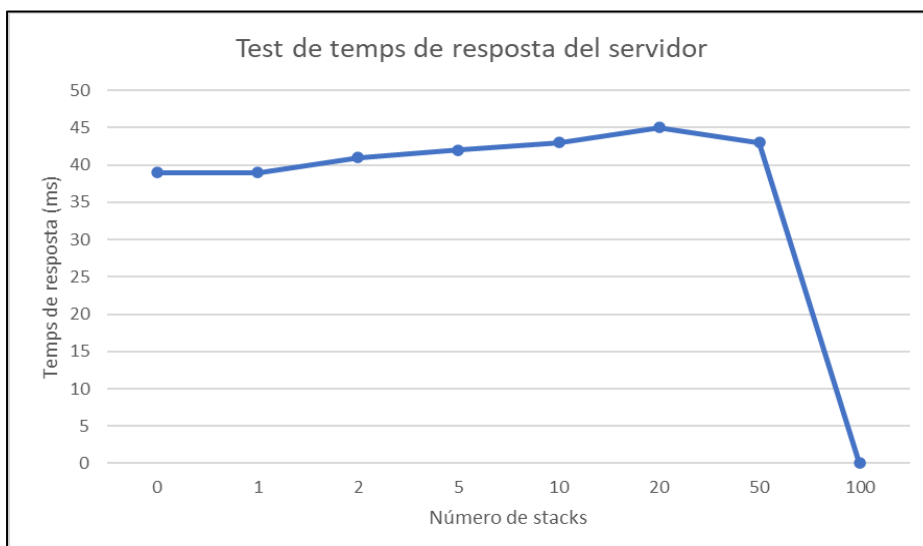
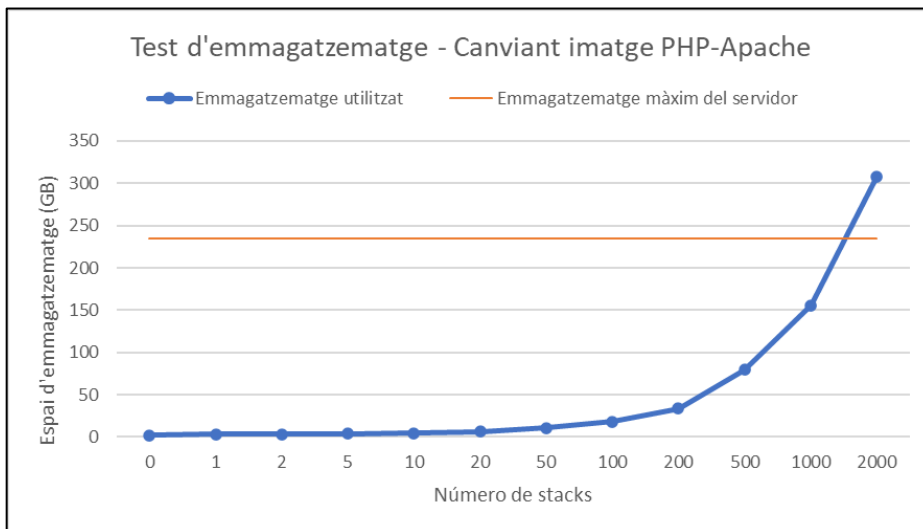
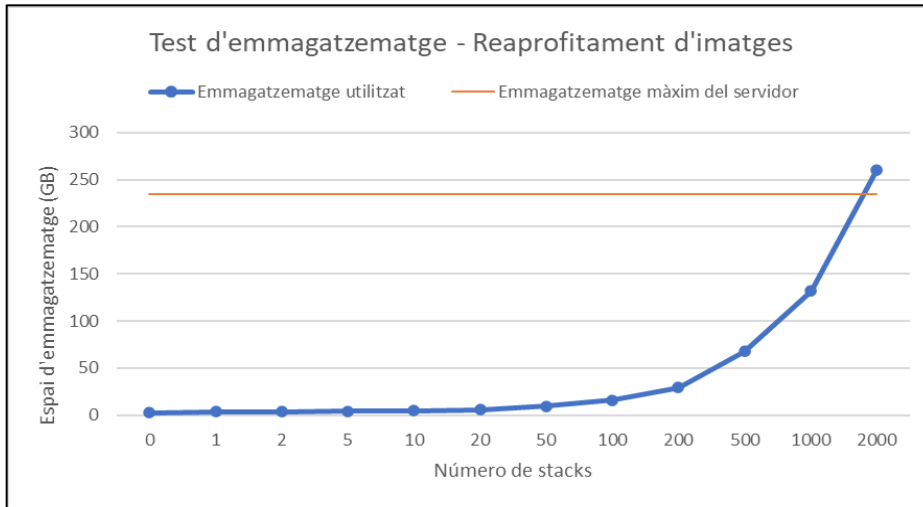
```

root@eng-informatica:~# ls /home/TFG/
tfg-b1  tfg-b2
root@eng-informatica:~# ls /home/TFG/tfg-b1
src
root@eng-informatica:~# ls /srv/TFG/
tfg-b1  tfg-b2
root@eng-informatica:~# ls /srv/TFG/tfg-b1
dades_stack.txt  Dockerfile  docker-stack.yml  mariadb
root@eng-informatica:~# ls /srv/TFG-produccio/
prova
root@eng-informatica:~# ls /srv/TFG-produccio/prova/
dades_stack.txt  Dockerfile  docker-stack.yml  mariadb  src

```

A3. Gràfics de les proves de rendiment

Proves d'emmagatzematge i de temps de resposta del servidor que no s'han mostrat anteriorment.



A4. Diagrama de Gantt de la planificació

Task Name	Duration	Start	Finish
Planificació	43 days	Mon 15/3/21	Tue 27/4/21
Planificació i definició de dates	7 days	Mon 15/3/21	Mon 22/3/21
Anàlisi de les eines actuals pel desenvolupament web	7 days	Mon 15/3/21	Mon 22/3/21
Recopilació de requisits funcionals i no funcionals	15 days	Mon 5/4/21	Mon 19/4/21
Realització d'un estudi de viabilitat tècnica	7 days	Tue 20/4/21	Tue 27/4/21
Realització d'un estudi ètic i legal	7 days	Tue 20/4/21	Mon 26/4/21
Reunió de definició del projecte	5 days	Tue 20/4/21	Sat 24/4/21
Elaboració de l'informe de progrés I	20 days	Mon 5/4/21	Sun 25/4/21
Entrega de l'informe de progrés I	0 days	Sun 25/4/21	Sun 25/4/21
Elaboració del projecte	36,88 days	Sun 25/4/21	Mon 31/5/21
Fase 1	10 days	Sun 25/4/21	Wed 5/5/21
Configuració de l'entorn de treball	3 days	Sun 25/4/21	Wed 28/4/21
Instal·lació i configuració del servidor Apache	4 days	Wed 28/4/21	Sun 2/5/21
Verificació del correcte funcionament de l'entorn de treball i servidor apache	2 days	Mon 3/5/21	Wed 5/5/21
Fase 2	13 days	Wed 5/5/21	Tue 18/5/21
Configuració del servidor per permetre l'inici de sessió amb el SSO de la UAB	6 days	Wed 5/5/21	Tue 11/5/21
Instal·lació i configuració de l'entorn Docker en el servidor	10 days	Thu 6/5/21	Sun 16/5/21
Verificació del correcte funcionament dels serveis web	2 days	Sun 16/5/21	Tue 18/5/21
Verificació del correcte funcionament de l'autenticació	2 days	Sun 16/5/21	Tue 18/5/21
Fase 3	36,88 days	Sun 25/4/21	Mon 31/5/21
Generació dels scripts i fitxers que permetin construir automàticament una pila de serveis web bàsica per a un alumne	34,88 days	Sun 25/4/21	Sat 29/5/21
Generació dels scripts i fitxers que permetin esborrar una pila de serveis web d'un alumne	7 days	Wed 19/5/21	Wed 26/5/21
Generació de documentació d'ajuda per a la creació de piles de serveis web personalitzades	32,88 days	Sun 25/4/21	Thu 27/5/21
Verificació del correcte funcionament dels scripts	2 days	Sun 30/5/21	Mon 31/5/21
Revisió de la documentació d'ajuda generada	2 days	Fri 28/5/21	Sat 29/5/21
Fase 4	6 days	Tue 18/5/21	Mon 24/5/21
Instal·lació i configuració del servei SSH	3,88 days	Tue 18/5/21	Sat 22/5/21
Verificació del correcte funcionament del servei SSH	2 days	Sat 22/5/21	Mon 24/5/21
Fase 5	6 days	Mon 24/5/21	Sun 30/5/21
Realització de tests de seguretat del servidor	6 days	Mon 24/5/21	Sun 30/5/21
Realització de tests d'estrès del servidor	6 days	Mon 24/5/21	Sun 30/5/21
Elaboració de l'informe de progrés II	34,88 days	Sun 25/4/21	Sat 29/5/21
Entrega de l'informe II	0 days	Sat 29/5/21	Sat 29/5/21
Tancament del projecte	44 days	Sat 29/5/21	Mon 12/7/21
Elaboració de l'informe final del projecte	14 days	Sat 29/5/21	Sat 12/6/21
Entrega de l'informe final	0 days	Sun 20/6/21	Sun 20/6/21
Elaboració de la presentació oral del projecte	14 days	Sat 12/6/21	Sat 26/6/21
Entrega presentació oral del projecte	0 days	Sun 27/6/21	Sun 27/6/21
Elaboració del dossier del projecte	2 days	Sat 26/6/21	Mon 28/6/21
Entrega del dossier	0 days	Mon 28/6/21	Mon 28/6/21
Elaboració del pòster del projecte	6,88 days	Mon 28/6/21	Sun 4/7/21
Entrega del pòster	0 days	Sun 4/7/21	Sun 4/7/21
Presentació oral i defensa del projecte	5 days	Mon 5/7/21	Fri 9/7/21
Tancament del TFG	0 days	Mon 12/7/21	Mon 12/7/21

