

Eina de suport al programador per desplegar una aplicació web amb Docker

Memòria de Projecte Final de Màster

Màster en Aplicacions Multimèdia

Itinerari professional

Autor: Joan Ricard Hernández Chaves

Consultor: Jordi Gervas Arruga

Professor: Laura Porta Simó

3 de gener del 2020

Copyright

CreativeCommons



Aquesta obra està subjecta a una llicència de Reconeixement

[3.0 Espanya de CreativeCommons](https://creativecommons.org/licenses/by/3.0/es/)

L'usuari és lliure de:

- **Compartir:** Copiar i redistribuir el material en qualsevol mitjà o format.
- **Adaptar:** Remesclar, transformar i construir a partir del material per qualsevol propòsit, inclús comercialment.

La llicenciant no pot revocar aquestes llibertats en tant vostè segueixi els termes de la llicència.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Eina de suport al programador per desplegar una aplicació web amb Docker</i>
Nom de l'autor:	<i>Joan Ricard Hernández Chaves</i>
Nom del consultor/a:	<i>Jordi Gervas Arruga</i>
Nom del PRA:	<i>Laura Porta Simó</i>
Data de lliurament:	<i>01/2020</i>
Titulació o programa:	<i>Màster en Aplicacions Multimèdia</i>
Àrea del Treball Final:	<i>Treball Final de Màster Professionalitzador</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Desplegament, Docker, Laravel</i>
Resum del Treball:	
<p>Al començar un nou projecte en el món del desenvolupament d'aplicacions web, per assolir els objectius proposats es fa servir una àmplia gama d'eines i recursos que existeixen per internet, com ara frameworks, plantilles, extensions, imatges o servidors que faciliten molt la vida al programador. Però realment no cal reinventar la roda quan ja existeixen recursos. Aquest projecte ve motivat perquè, per experiència, al començar nous projectes el que es vol és desenvolupar amb la tecnologia més actual i buscar quines eines o recursos són els adequats pel nostre cas. No obstant, pot arribar un cert moment que per cada nou projecte que s'inicia repeteix el mateix procés que l'anterior: recreació de la mateixa infraestructura de servidors, mateixa tecnologia de Front-end o Back-end i la nova o mateixa plantilla web.</p> <p>Llavors és ara quan sorgeix la pregunta de: I per què no tindre a l'abast un paquet de recursos predeterminats amb un servidor i una plantilla web configurat i preparat amb tot el necessari per muntar una aplicació que fos reutilitzable, escalable en el temps i a punt per ser publicat?</p> <p>Aquest Treball Final ha tingut com a finalitat, amb èxit i en una fase inicial, facilitar a la figura del programador/a la possibilitat d'executar un script on faci automàticament una sèrie d'instal·lacions que va des de la configuració d'un servidor preparat amb tecnologia Docker fins a la publicació d'una plana web definida i feta en Laravel, Bootstrap i jQuery amb el seu BackOffice amb possibilitat d'ampliar i afegir més funcionalitats.</p>	
Abstract:	
<p>When starting a new project in the world of web application development, to achieve the objectives proposed is to serve a wide range of tools and resources that exist on the Internet, such as frameworks, templates, plugins, images or servers that facilitate the programmer's life. Indeed, we really don't want to reinvent the wheel when resources already exist. This project is motivated by our own experience, because when we start our projects we want to develop with the latest technology and find out which resources are appropriate for our project.</p> <p>However, we can arrive at a certain moment that for each new project that begins is a repetition of the</p>	

same process as the previous one: recreation of the same servers infrastructure, the same front-end and back-end technology and the new or the same web template.

Now is when the question arises: And why not have a pre-configured resource pack with a prepared server and a web template with all the necessary to build a reusable and scalable application over time and ready to be published?

This Final Master's Project has had purpose of, successfully and an initial phase, facilitate to the programmer to given the possibility of executing a script on a series of installations which go from the configuration of a server prepared with Docker technology to the publication of a defined and ready-made web page in Laravel, Bootstrap and jQuery with their BackOffice with the possibility of expanding and adding more functions.

Agraïments

Agrair al professor de l'assignatura Jordi Gervas Arruga per la seva disponibilitat i interès davant de qualsevol problema o dubte que sorgís durant el desenvolupament del projecte.

Resum del Treball

Al començar un nou projecte en el món del desenvolupament d'aplicacions web, per assolir els objectius proposats es fa servir una àmplia gama d'eines i recursos que existeixen per internet, com ara frameworks, plantilles, extensions, imatges o servidors que faciliten molt la vida al programador. Però realment no cal reinventar la roda quan ja existeixen recursos. Aquest projecte ve motivat perquè, per experiència, al començar nous projectes el que es vol és desenvolupar amb la tecnologia més actual i buscar quines eines o recursos són els adequats pel nostre cas. No obstant, pot arribar un cert moment que per cada "següent projecte" que comença es repeteixi el mateix procés que l'anterior: recreació de la mateixa infraestructura de servidors, mateixa tecnologia de Front-end o Back-end i la nova o mateixa plantilla web.

Lavors és ara quan sorgeix la pregunta de: I per què no tindre a l'abast un paquet de recursos predeterminats amb un servidor i una plantilla web configurat i preparat amb tot el necessari per muntar una aplicació que fos reutilitzable, escalable en el temps i a punt per ser publicat?

Aquest Treball Final ha tingut com a finalitat, amb èxit i en una fase inicial, facilitar a la figura del programador/a la possibilitat d'executar un script on faci automàticament una sèrie d'instal·lacions que va des de la configuració d'un servidor preparat amb tecnologia Docker fins a la publicació d'una plana web definida i feta en Laravel, Bootstrap i jQuery amb el seu BackOffice amb possibilitat d'ampliar i afegir més funcionalitats.

Abstract

When starting a new project in the world of web application development, to achieve the objectives proposed is to serve a wide range of tools and resources that exist on the Internet, such as frameworks, templates, plugins, images or servers that facilitate the programmer's life. Indeed, we really don't want to reinvent the wheel when resources already exist. This project is motivated by our own experience, because when we start our projects we want to develop with the latest technology and find out which resources are appropriate for our project.

However, we can arrive at a certain moment that for each new project that begins is a repetition of the same process as the previous one: recreation of the same servers infrastructure, the same front-end and back-end technology and the new or the same web template.

Now is when the question arises: And why not have a pre-configured resource pack with a prepared server and a web template with all the necessary to build a reusable and scalable application over time and ready to be published?

This Final Master's Project has had purpose of, successfully and an initial phase, facilitate to the programmer to given the possibility of executing a script on a series of installations which go from the configuration of a server prepared with Docker technology to the publication of a defined and ready-made web page in Laravel, Bootstrap and jQuery with their BackOffice with the possibility of expanding and adding more functions.

Paraules clau

Eina, Desplegament, Programador, Laravel, Bootstrap, Docker, Aplicació Web.

Keywords

Tool, Deployment, Programmer, Laravel, Bootstrap, Docker, Web Application.

Índex

Capítol 1: Introducció	11
1. Introducció	11
2. Descripció	12
3. Objectius generals.....	13
3.1 Objectius principals	13
3.2 Abast	14
4. Metodologia i procés de treball	15
5. Planificació.....	16
6. Pressupost.....	17
7. Estructura de la resta del document	18
Capítol 2: Anàlisi	19
1. Estat de l'art	19
2. Públic objectiu i perfils d'usuari	22
3. Definició d'objectius i especificacions del producte	23
Capítol 3: Disseny	24
1. Arquitectura general de l'aplicació i del sistema	24
2. Arquitectura de la informació i diagrames de navegació	25
3. Disseny gràfic i interfícies	27
3.1 Estils	27
3.2 Usabilitat i UX	27
4. Llenguatges de programació.....	28
Capítol 4: Implementació	30
1. Requisits d'instal·lació	30
2. Instruccions d'instal·lació	31
Capítol 5: Demostració	32
1. Instruccions d'ús.....	32
2. Tests.....	33
4. Exemples d'ús del producte.....	35
Capítol 6: Conclusions i línies de futur	36
1. Conclusions.....	36
2. Línies de futur	37

Bibliografia	38
Annexos	40
Annex A: Glossari	40
Annex B: Lliurables del projecte	40
Annex C: Captures de pantalla	41

Figures i taules

Índex de figures

Figura 1: Framework Bootstrap i Vue.js	19
Figura 2: Framework de PHP Laravel	20
Figura 3: Programari lliure de virtualització VirtualBox	21
Figura 4: Docker, tecnologia d'aplicacions en contenidors	21
Figura 5: Taules de la base de dades	25
Figura 6: Diagrama de navegació de la plana web (Front-end)	25
Figura 7: Diagrama de navegació del panell d'administració (BackOffice)	26
Figura 8: Plantilles bàsiques de Bootstrap	27
Figura 9: Resultat final del procés d'instal·lació de l'script	32
Figura 10: Creació dels contenidors Docker	33
Figura 11: Contenidors Docker funcionant	33
Figura 12: Contingut de l'arxiu hosts	33
Figura 13: Captura del contingut del front-end	34
Figura 14: Captura del contingut del backoffice	34
Figura 15: Comandes de servidor en bash	35
Figura 16: Estructura del servidor virtual	35
Figura 17: Trello, eina de control del desenvolupament	41
Figura 18: Front-end – Pàgina de portada	42
Figura 19: Front-end – Pàgina sobre nosaltres	43
Figura 20: Front-end – Pàgina de blog	44
Figura 21: Front-end – Pàgina de contacte	45
Figura 22: BackOffice – Pàgina de Login	46
Figura 23: BackOffice – Pàgina de registre d'usuari	46
Figura 24: BackOffice – Pàgina de contactes	46
Figura 25: BackOffice – Pàgina d'informació genèrica	47
Figura 26: BackOffice – Pàgina de blog: Modificar contingut	47
Figura 27: BackOffice – Pàgina de blog: Nou contingut	48

Índex de taules

Taula 1: Fites del desenvolupament del projecte	16
---	----

Capítol 1: Introducció

1.Introducció

En l'àmbit del desenvolupament d'aplicacions web, moltes vegades en el moment de començar un nou projecte, per tal d'assolir els objectius proposats, es fan servir una àmplia gama d'eines i recursos que es pot obtenir per internet, com ara frameworks, plantilles, plugins, imatges o servidors web que faciliten molt la vida al programador i que realment no cal reinventar la roda quan ja existeixen recursos preparats per fer-los servir.

Aquest projecte ve motivat perquè, per experiència personal i professional, al començar nous projectes el que es vol és desenvolupar amb la tecnologia més puntera i s'ha de buscar quines eines i recursos són les més adequades pel nostre propòsit. A més a més, pot arribar un cert moment que per a cada nou projecte que es comença de zero, es repeteix una i altra vegada el mateix procés que l'anterior: Recreació de la mateixa infraestructura de servidors, mateixa tecnologia de Front-end i de Back-end i mateixa plantilla web.

Per aquesta raó és quan ara sorgeix la pregunta de: "I per què no tindre a l'abast un paquet de recursos on amb un servidor web ja configurat i una plantilla web ben preparada amb tot el necessari per muntar una aplicació que fos reutilitzable i escalable en el temps?" Com a programador web habitual, s'ha arribat a aquesta qüestió, i per facilitar als companys amb la mateixa inquietud, sorgeix la necessitat de crear tot un paquet de recursos configurats amb antelació i poder-ho desplegar en poc temps, muntar i donar servei a una aplicació web de la manera més ràpida i senzilla mitjançant uns paràmetres predefinits. El programador que el farà servir el podrà instal·lar i modificar al seu gust, i si cal en el mateix moment o en un futur, ampliar-lo amb més funcionalitats. Per aquest treball es seleccionen tecnologies de codi obert amb els frameworks que poden ser més habituals, útils i més agradables possibles en aquest moment i que es detallen durant aquest document amb la possibilitat de canviar per altres o modificar les existents. Gran part de la motivació d'haver fet aquest projecte és el fet de que es pot reutilitzar molt i per la varietat de tecnologies en el que es pot combinar.

Partint d'aquesta necessitat, es crea una eina de suport al programador per desplegar una aplicació web amb la tecnologia Docker[1], capaç de ser integrat des de qualsevol servidor proveïnt d'una aplicació web, amb la seva part visual, part de panell d'administració, base de dades i servidor web. El propòsit del present TFM és desenvolupar i configurar cada part esmentada i que mitjançant un fitxer script pugui ser capaç de fer tota la instal·lació tant desatesa com sigui possible.

2. Descripció

El món de les aplicacions web és molt ampli i existeixen múltiples tecnologies que es poden fer servir a l'hora de desenvolupar un nou lloc web. Pel que fa al Front-end es poden trobar tecnologies com HTML5[2], CSS3[3], JavaScript[4], frameworks de caràcter visual com Bootstrap[5], Material Design[6] o Materialize[7] i frameworks més complexes basats en JavaScript com Angular[8], React[9] o Vue[10]. Per altra banda, pel que fa al Back-end es poden trobar tecnologies com PHP[11] i frameworks tipus Laravel[12] o Django[13]. Pel que fa a la persistència de dades, entre altres, existeix MySQL[14] / MariaDB[15], MongoDB[16] o PostgreSQL[17] i moltes altres tecnologies on les opcions són inesgotables. El primer que cal determinar és quina tecnologia pot ser la més adequada per un projecte i en aquest cas s'han agafat aquelles que poden ser més usuals i una corba d'aprenentatge que sigui més fàcil.

Un dels punts forts d'aquest TFM, és l'ús de la tecnologia Docker, que ha entrat amb molta força (i que no deixa indiferent a ningú) per canviar el paradigma sobre la forma en la que es desenvolupa, es distribueix i executa el software als servidors encapsulant tot l'entorn de treball en forma de contenidors[18], executant-la en qualsevol entorn i exportant-la sempre amb la mateixa configuració. Està molt orientat sobretot per a administradors de sistemes i desenvolupadors. Aporta gran utilitat al crear contenidors lleugers, independents entre ells i portables, ja que poden executar-se des de qualsevol servidor independentment del sistema operatiu. La gran utilitat de Docker és que, si fins ara el que es fa és crear multitud de màquines virtuals per publicar diferents aplicacions, ara només es cal crear una màquina virtual amb multitud de contenidors reaprofitant els recursos del Sistema Operatiu com a base comuna.

La necessitat que es cobreix amb l'eina o paquet de recursos, és brindar ajuda al programador a obtenir una base amb tot el necessari que s'ha de tindre per muntar una aplicació web, començant des de la instal·lació d'un servidor web fins la plantilla d'una pàgina amb l'objectiu de no perdre temps en configurar i programar. Enfocar la idea d'aquest treball final aporta valor a la feina del programador pel volum de projectes que poden demanar i que tindre ja preparat i configurat un paquet d'aquest estil estalvia molta feina repetitiva, temps de dedicació i permet tindre un software actualitzat i punter preparat per ser instal·lat. Així doncs, la intenció és executar un sol arxiu que conté totes les comandes necessàries per muntar tota la estructura de forma desatesa.

Amb el Treball Final de Màster es pretén crear una fase inicial d'una eina pública de suport al programador que servirà com a base amb la intenció de fer una recopilació de recursos per muntar i donar servei a una aplicació web, allotjada en un repositori des de GitHub i desplegada de forma automàtica mitjançant un script, on s'anirà ampliant en el temps amb més funcionalitats, millors configuracions i endinsar-se de ple cap a la tecnologia Docker.

3. Objectius generals

El present Treball Final de Màster està orientat cap al perfil del desenvolupador d'aplicacions web. Principalment l'objectiu d'aquest projecte és implementar una plataforma capaç d'ajudar al programador a tenir a l'abast un paquet de recursos suficients com per poder muntar i tindre preparat i configurat un servidor i una aplicació web implementat en molt poc temps. Un cop instal·lat, el programador podrà ampliar o modificar aquests recursos tant com vulgui.

3.1 Objectius principals

Instal·lar i configurar manualment un servidor web amb Docker:

- Investigar el funcionament de Docker.
- Desplegar tres contenidors que faran funcionar les plantilles web:
 - Crear i configurar contenidor amb el servidor web Apache2[19].
 - Crear i configurar contenidor amb el servidor de bases de dades MySQL/MariaDB.
 - Crear i configurar contenidor amb PHP.

Desenvolupament de plantilles web pel front-end:

- Implementar una plantilla web mitjançant el framework Laravel i Bootstrap.
- Implementar la funcionalitat de les plantilles mitjançant el framework JavaScript jQuery i AJAX[20] en base d'unes plantilles bàsiques gratuïtes sota llicència MIT d' StartBootstrap[21].
- Connectar amb la base de dades MySQL.

Desenvolupament d'una plantilla web en funció del panell d'administració:

- Implementar una plantilla web mitjançant el framework Laravel, Bootstrap i React, en base a una plantilla gratuïta sota llicència MIT anomenat: AdminLTE[22].
- Implementar funcionalitats per modificar les dades de la base de dades MySQL.

Configuració i automatització de comandes necessàries per la instal·lació de Docker i plantilles web:

- Investigar com crear un procés / script per automatitzar tot el procés d'instal·lació i configuració dels contenidors Docker.

Objectius per al usuari:

- Executar l'script creat prèviament per la instal·lació de tot el software necessari per desplegar i instal·lar un servidor web amb les seves plantilles web de forma desatesa.

Objectius personals de l'autor del TF:

- Investigar i aprofundir tan com sigui possible sobre la tecnologia Docker.
- Generar una eina d'ajuda al programador per desplegar aplicacions web mitjançant un servidor i una plantilla predefinida.

- Elaborar una memòria de TFM que inclogui tot el procés de desenvolupament i anàlisi realitzada de forma prèvia.
- Aplicar els coneixements que s'han adquirit durant l'estudi del Màster i adquirir experiència en la planificació d'un projecte des de zero.
- Ampliar els coneixements sobre diferents tecnologies web que encara no s'han tractat.

3.2 Abast

El producte que s'ha d'entregar al final del TFM ha de contenir un fitxer script completament funcional amb les comandes necessàries per poder-lo executar i que faci tot el desplegament del software que farà funcionar i donarà servei a les plantilles de les pàgines web.

Apart, tant les plantilles web desenvolupades com els recursos necessaris per la configuració dels contenidors Docker, han de ser allotjades en un repositori públic publicat a la pàgina web de GitHub, amb la intenció de que sigui descarregable per qualsevol usuari interessat.

A més a més, tots els recursos que es publiquen són sota llicència MIT, que permet a qualsevol usuari poder descarregar, modificar, publicar i distribuir tant com vulgui.

Finalment, caldrà incloure també una presentació acadèmica i una presentació pública amb l'objectiu de fer saber al tribunal i al públic la finalitat d'aquest projecte.

4. Metodologia i procés de treball

La metodologia de desenvolupament que s'ha fet servir per aquest projecte, pertany al model de metodologia àgil Scrum[23], que consisteix en fixar fites d'entre 1 a 4 setmanes amb l'objectiu d'implementar, de revisar i d'obtenir les funcionalitats previstes dins de les dates proposades a la planificació i, si cal, modificar-la i adequar-la a les noves decisions.

Per portar a cap una bitàcola de control de totes les tasques per fer i de les finalitzades, s'ha fet servir una eina externa gratuïta anomenada "Trello"[24], on mitjançant la metodologia Kanban[25] es van definir per a cada tasca diferents columnes d'estat entre totes les fites:

- **PENDENT**: Llistat de tasques pendents que s'han de desenvolupar.
- **EN PROCÉS**: Per aquelles tasques que actualment s'estan desenvolupant.
- **FINALITZAT**: Per aquelles tasques que ja s'han realitzat.
- **TASQUES PERMANENTS**: Per aquelles tasques que perdura en tot el projecte.
- **IDEES INICIALS DESCARTADES**: Donats uns objectius inicials, poden haver canvis durant el desenvolupament. Per aquest cas, s'ha creat un tauler on deixar un llistat de canvis.

De cara a desenvolupar les plantilles web, s'ha fet servir recursos externs des de "StartBootstrap" publicats sota llicència MIT amb la finalitat d'aprofitar els exemples de recursos ja existents exemplificats per Bootstrap i aplicats i emmotllats a aquest projecte. Per altre costat, pel que fa als contenidors Docker, un cop s'ha fet la investigació sobre el funcionament i la configuració dels contenidors, s'ha fet servir un paquet de recursos externs publicats sota llicència MIT anomenat "Laradock"[26] amb la finalitat d'aprofitar-los i configurar-los per l'objectiu previst en aquest treball final.

El desenvolupament del projecte s'ha dividit a partir de les següents quatre fases, considerats en l'ordre de igual importància amb la finalitat de començar a aprendre sobre la tecnologia Docker i programar les plantilles web en base al que s'ha après:

1. Investigar, instal·lar i configurar un servidor web amb Docker per a cada una de les tecnologies utilitzades: MySQL, PHP7.2 i Apache2.
2. Agafar idees de recursos visuals gratuïts fets per Bootstrap i dissenyar i implementar la base de dades on es col·loca la persistència de la informació.
3. Desenvolupament i testejar tant la pàgina web frontal (Front-end) com la del panell d'administració (BackOffice) amb el framework de PHP Laravel versió 6.
4. Configuració del fitxer de comandes Docker: Fer el desplegament automàtic generant tota l'estructura de contenidors per fer funcionar les dues aplicacions web.

Aquesta ha estat l'estratègia més apropiada considerant els recursos dels que es disposaven: Un únic desenvolupador amb una línia de temps molt limitat i posant més pes sobre la investigació de Docker i de l'automatització del desplegament que del desenvolupament de les pàgines web.

5. Planificació

Pel desenvolupament d'aquest projecte, es defineixen les fites importants a partir de les dates de les PAC com a part del producte que ha d'estar enllestit a cada entrega parcial i organitzat d'acord amb el calendari acadèmic:

Tasques i fites del desenvolupament del projecte			
Nom	Durada (dies)	Inici	Final
PAC2	10	05/10/2019	14/10/2019
Inici de l'informe inicial amb la redacció de l'estat de l'art i recerca d'informació.	3	05/10/2019	07/10/2019
Redacció d'objectius i abast del projecte.	3	08/10/2019	10/10/2019
Planificació de les fites i temps de dedicació.	2	11/10/2019	12/10/2019
Correccions de l'informe inicial.	2	13/10/2019	14/10/2019
PAC3	28	15/10/2019	11/11/2019
Creació i configuració del servidor.	1	15/10/2019	15/10/2019
Instal·lació i configuració inicial de Docker.	1	15/10/2019	15/10/2019
Instal·lació i configuració de Laravel 6.	1	16/10/2019	16/10/2019
Disseny inicial de la plantilla de la pàgina web.	2	17/10/2019	18/10/2019
Disseny inicial del panell d'administrador.	2	19/10/2019	20/10/2019
Disseny i implementació de la base de dades.	2	21/10/2019	22/10/2019
Implementació de la plantilla de la pàgina web.	10	23/10/2019	31/10/2019
Connectar a la BD la plantilla de la pàgina web i Implementació del panell d'administració.	11	01/11/2019	11/11/2019
PAC4	28	12/11/2019	09/12/2019
Implementació del panell d'administració.	5	12/11/2019	16/11/2019
Proves de les implementacions front i back.	2	17/11/2019	18/11/2019
Automatització dels contenidors Docker.	15	19/11/2019	02/12/2019
Proves d'execució dels contenidors Docker.	7	03/12/2019	09/12/2019
PAC5	25	10/12/2019	03/01/2020
Últims retocs del desenvolupament del projecte.	6	10/12/2019	15/12/2019
Redacció i correcció de la memòria.	12	16/12/2019	27/12/2019
Elaboració de la presentació.	7	28/12/2019	03/01/2020
PAC3 + PAC4 + PAC5	81	15/10/2019	03/01/2020
Proves d'implementació i elaboració de documents.	81	15/10/2019	03/01/2020

Taula 1: Fites del desenvolupament del projecte

6. Pressupost

Per la realització d'aquest projecte no ha calgut preveure de cap tipus de pressupost, ja que tots els recursos que s'han fet servir pel muntatge de l'eina de suport al programador han sigut amb programari lliure. Així mateix, pel que fa a tots els recursos web estan lligats sota llicència MIT, amb el dret de fer servir, copiar, modificar, fusionar, distribuir, sublllicenciar, vendre còpies, i amb la mateixa dinàmica, aquest projecte hereta la mateixa trajectòria per tot aquell programador que el vulgui fer servir amb total llibertat.

Els recursos gratuïts que s'han fet servir han sigut:

Software de virtualització

- **VirtualBox d'Oracle[27]**: Programari de virtualització de sistemes operatius.
- **Sistema Operatiu[28]**: Ubuntu Server versió 18.04 LTS.

Software de programació

- **Visual Studio Code[29]**: Programari de desenvolupament web fet per Microsoft.
- **SublimeText3[30]**: Programari de desenvolupament amb opció de pagar llicència si es precisa.

Recursos web sota llicència MIT

- **StartBootstrap**: Plantilles senzilles basades en el framework Bootstrap.
- **Docker**: Tecnologia de servidor basat en contenidors.
- **Laradock**: Paquet de recursos Docker dedicat a la tecnologia Laravel.
- **Laravel**: Framework web de PHP.
- **Font Awesome 5[31]**: Paquet de recursos d'icones.

Imatges sense copyright

- **Pexels.com[32]**: Pàgina web amb imatges gratuïtes de fer servir com i quan es vulgui sense que sigui necessari nombrar l'origen del recurs.

Per altre costat, el cost de mà d'obra depèn de les intencions del programador/a que l'estigui fent servir:

- Cas d'un programador/a amb fins no comercials: Cost assumit.
- Cas d'un programador/a amb fins comercials: Cost per hores o de producte tancat.

7. Estructura de la resta del document

Breu descripció dels altres capítols de la Memòria:

Capítol 2: Anàlisi

- **Estat de l'art:** Descripció en detall de quin és l'estat de l'art sobre les tecnologies web que existeixen i que venen de noves per quedar-se, programari que els complementa i tipus de llicències que sustenta el mercat.
- **Públic objectiu i perfil d'usuari:** Breu descripció de quin perfil d'usuari es dirigeix aquest TFM.
- **Definició d'objectius i especificacions del producte:** Breu descripció del què ha de fer aquests recursos al final d'aquest treball i què s'hauria d'aconseguir amb l'eina de suport al programador.

Capítol 3: Disseny

- **Arquitectura general de l'aplicació i del sistema:** Descripció general dels elements que componen el producte desenvolupat, l'aplicació i la forma en què es relacionen entre sí.
- **Arquitectura de la informació i diagrames de navegació:** Descripció detallada mitjançant diagrames dels elements principals que componen el producte.
- **Disseny gràfic i interfícies:** Breu descripció del procés de disseny de les aplicacions.
- **Llenguatges de programació:** Breu descripció dels llenguatges de programació que s'han fet servir durant el desenvolupament.

Capítol 4: Implementació

- **Requisits d'instal·lació:** Explicació de quins són els requisits en el cas de ser necessari una instal·lació.
- **Instruccions d'instal·lació:** En cas de ser necessari, explicació dels passos detallats per instal·lar el programari d'aquest projecte.

Capítol 5: Demostració

- **Instruccions d'us:** Descripció dels passos que s'han de realitzar per la utilització del recurs.
- **Tests:** Proves realitzades a la posada en marxa dels recursos, funcionalitat i utilitat.

Capítol 6: Conclusions i línies de futur

- Conclusió personal sobre el projecte realitzat, procés de treball, resultats i el futur del mateix.

Capítol 2: Anàlisi

1. Estat de l'art

A vegades, molts dels programadors a l'hora de desenvolupar els seus projectes, busquen la tecnologia adient per assolir els objectius inicialment proposats. En molts casos, pel que fa a la tecnologia web, per facilitar la feina o simplement per estalviar temps se solen buscar diferents plantilles web que satisfaci les necessitats:

- Existeixen exemples de **pàgines web estàtiques** (no connectades a cap base de dades) en HTML5 i CSS3 ja programades i moltes vegades porten alguns afegits amb plugins en jQuery o similars que són fetes per empreses privades o programadors autònoms on inclou tot un catàleg d'exemples ideals per cobrir tot el necessari i preocupar-se mínimament del disseny de la pàgina sense ser un dissenyador professional.
- Per altre costat i igual que l'anterior, hi ha exemples de pàgines fetes però amb l'objectiu d'implementar un **"panell d'administració / BackOffice"** per poder mostrar i administrar visualment les dades d'una pàgina web, amb formularis o graelles per gestionar tota la informació allotjada a la base de dades. Ja que són uns exemples, en un futur s'ha d'adaptar i crear una connexió amb una base de dades que finalment vindrà lligat amb la pàgina web.

Actualment en el mercat existeixen moltes plantilles fetes amb una bona varietat de tecnologies, com per exemple:

- Pel que fa al **"panell d'administració / BackOffice"**, existeix Bootstrap, una eina de codi obert molt popular que ofereix a l'usuari construir una plana web pas per pas basat en HTML5, CSS3 i JavaScript que ofereix moltes utilitats relacionades amb la interfície d'usuari. D'aquest, existeixen populars varietats d'estils creats per altres entitats, com: Bootstrap[33] amb Angular, React i Vue o Bootstrap amb Material Design[34], dissenyat per Google i d'aquest[35] complementat amb Angular, React i Vue[36]. Es poden trobar variants com Materialize, un framework amb un disseny modern, adaptat per a qualsevol dispositiu molt similar al Bootstrap però amb marca pròpia i basat amb Material Design.



Figura 1: Framework Bootstrap i Vue.js

- Pel que fa a una **pàgina web**, existeixen multituds de pàgines que ofereixen planes web amb tecnologies bassades en frameworks com Bootstrap, React, Angular, Laravel o simplement HTML5 i CSS3 propis.



Figura 2: Framework de PHP Laravel.

Aquestes plantilles web generalment estan associades sota a unes llicències de propietat intel·lectual per evitar frau innecessaris:

1. **Llicència MIT:** Amb permís sense càrrec a qualsevol persona que obtingui una còpia del software sense restricció amb els drets d'usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar i / o vendre còpies del software sota la condició de:
 - a. La còpia del software s'obté de forma íntegra.
 - b. Sense dret a cap tipus de reclamació, danys o altres responsabilitats com una acció de contracte, tortura o d'altres maneres i fora o en connexió amb el software amb l'ús o d'altres ofertes.
2. **Llicència Creative Commons:** Aquest dona permís a l'usuari de poder fer servir la plantilla, copiar, redistribuir el material en qualsevol medi o format, adaptar o transformar i construir a partir del material per qualsevol propòsit incloent comercialment amb la condició de que es col·loqui la referència de l'autor, és a dir, compartir des d'on s'ha adquirit el producte.
3. **Llicència de pagament:** En aquest cas, l'usuari paga per una plantilla web i es pot derivar a dos tipus de condicions:
 - a. **Pagar una plantilla per un únic ús:** Per aquest primer cas es té dret a tot excepte que la plantilla es publica únicament una sola vegada, és a dir, es pot utilitzar un cop. En cas de voler crear una altra pàgina web amb la mateixa plantilla s'ha de tornar a pagar.
 - b. **Pagar una plantilla per múltiples usos:** Per aquest segons cas es té dret a tot i es pot reutilitzar múltiples vegades.

Ara bé, per poder allotjar i publicar aquestes pàgines bassades en HTML, CSS i JavaScript, el que es necessita és obtenir i configurar un servidor amb un Sistema Operatiu Windows o Linux i instal·lar tot el software necessari. Actualment en el mercat existeixen tota una varietat de software:

- **Servidors web** com APACHE2, NGINX[37], Tomcat[38] o Glassfish[39].
- Tecnologies de **Back-end** com PHP, Python[40] o JavaEE[41].

- Servidors de **base de dades** de codi obert com MySQL, MariaDB, MongoDB o Postgree i privats amb compra de llicència com SQL[42].

Aquest servidors, poden ser de dos tipus:

- **Servidors físics:** Aquest és un ordinador especialment dissenyat per realitzar funcions concretes i que funciona sota uns requisits específics. A més, comporta un cost important donat la necessitat d'adquirir altres dispositius com routers, SAIS, generadors o switches, sistemes de refrigeració, sistemes de seguretat, extinció de focs, espai o electricitat.
- **Servidors virtuals:** Aquest són els que s'instal·len mitjançant software i que simulen un servidor físic. A diferència de l'anterior, aquest no presenta de cap cost addicional sinó el d'una llicència de virtualització si es d'una marca privada com Citrix[43] i VMWARE[44] o de gratuïts com VirtualBox d'Oracle.



Figura 3: Programari lliure de virtualització VirtualBox

Donat la varietat de servidors virtuals que existeixen, actualment la majoria de les aplicacions web que hi ha per Internet són allotjades i distribuïdes sobre servidors virtuals anomenats VPS (Servidor Virtual Privat) amb tota la configuració anomenada anteriorment per poder-les executar i distribuïdes per empreses dedicades a aquest servei. Fins ara, la relació que hi ha entre servidor i aplicació és d'un en un, és a dir: 1 servidor = 1 aplicació web.

Al març del 2013 va sorgir un projecte de codi obert anomenat Docker amb la intenció d'automatitzar el desplegament d'aplicacions dintre de contenidors de software, proporcionant una capa addicional d'abstracció i automatització de virtualització d'aplicacions en molts sistemes operatius fent que aquest sigui reutilitzable. L'objectiu principal de Docker és estalviar recursos sobre aquest servidors, evitant crear el que és 1 servidor = 1 aplicació web. Donat que la base de cada servidor és el sistema operatiu, Docker aprofita aquest recursos per generar contenidors com si fossin "mini servidors virtuals" amb la diferència que només es necessita un únic servidor per muntar tants aplicacions com es vulgui.



Figura 4: Docker, tecnologia d'aplicacions en contenidors

2. Públic objectiu i perfils d'usuari

El públic objectiu d'aquest projecte ve enfocat directament per aquells programadors de tots els perfils, edat i amb diversa experiència que desenvolupen o volen desenvolupar aplicacions web.

Aquestes persones poden identificar-se clarament amb la necessitat de tindre a l'abast un paquet de recursos amb la intenció d'estalviar temps en la creació de les seves aplicacions.

3. Definició d'objectius i especificacions del producte

L'eina de suport al programador ha d'instal·lar automàticament tot el software necessari per que funcioni la tecnologia Docker i que configuri els contenidors d'Apache2, MySQL i PHP perquè quan descarregui i instal·li les plantilles web, creats amb tecnologia de PHP amb Laravel versió 6, des del repositori GitHub i aquestes puguin ser accessibles mitjançant una URL en el moment que finalitzi tot el procés.

Amb la tecnologia Docker, aquest recursos poden ser instal·lats a qualsevol servidor sense que es vegi compromès a problemes de versions de maquinari i programari.

Pel que fa a les plantilles web, ha de ser capaç de tindre un mínim de continguts i funcionalitats perquè sigui funcional des del moment de finalitzar el projecte:

1. **Pàgina web:** Plantilla web front-end que ha de mostrar la següent informació:
 - Per defecte ha de mostrar les pàgines de portada, sobre nosaltres i equip professional, blog, mapa i formulari de contacte que enviarà les dades cap a la base de dades i es mostraran al panell d'administració.
2. **Panell d'administració:** Ha d'aparèixer tota la informació de la base de dades. Aquestes han de ser modificables i tanmateix poder eliminar i registrar més dades. Per tant:
 - Ha de comunicar-se amb la base de dades.
 - Ha d'afegir, mostrar i eliminar les dades de la "portada".
 - Ha d'afegir, mostrar i eliminar les dades de "sobre nosaltres i equip professional".
 - Ha d'afegir, mostrar i eliminar les dades del formulari de "contacte".
 - Ha d'afegir, mostrar i eliminar entrades de "blog".
3. **Base de dades:** La base de dades ha de crear-se i configurar l'usuari i la contrasenya per defecte "root" "root" respectivament.

Pel que fa al repositori GitHub on es guarden les plantilles i els recursos de configuració dels contenidors Docker, és:

- <https://github.com/jrhernandezch>

A més, amb aquest repositori també es pot fer servir com a lloc segur on fer còpia de seguretat del codi. Durant el desenvolupament, cada repositori queda registrat de manera privada mentre que posteriorment, al finalitzar el període de desenvolupament, es fa públic per que qualsevol usuari que el vulgui fer servir, el pugui descarregar i executar.

Capítol 3: Disseny

1. Arquitectura general de l'aplicació i del sistema

El patró d'arquitectura que s'ha fet servir en el desenvolupament de les planes web és el de Model – Vista - Controlador (MVC) mitjançant el framework Laravel. Aquest és un model que separa la interfície d'usuari i la lògica de negoci i es basa en la idea de reutilitzar i escalar el codi.

És a dir:

- **Vista:** És la interfície gràfica que veu l'usuari i mostra la informació que el controlador proporciona a la vista.
- **Controlador:** És la capa intermèdia entre la vista i el model. Fa la funció de les peticions que l'usuari demana al interactuar amb la vista. Per tant, l'usuari demana informació i el controlador es comunica amb el model per demanar el que l'usuari vol i rebre les dades cap a la vista.
- **Model:** És la part encarregada de gestionar la persistència de les dades. Gestiona els accessos a la informació mitjançant consultes i actualitzacions amb la base de dades.

2. Arquitectura de la informació i diagrames de navegació

Per preservar les dades de la plantilla web, es va crear una base de dades amb MariaDB / MySQL on poder guardar tota la informació que pot anar canviant mitjançant el panell d'administració i que anirà lligat amb la plantilla de la pàgina web. Durant el seu desenvolupament, s'ha anat canviant i finalment ha quedat de la següent manera:

Table Name	Columns
laravel_business professional_team	id_team: int(11), name: varchar(150), position: varchar(50), content: text, mail: varchar(100), image: varchar(150), date: timestamp
laravel_business contact_info	id_contact_info: int(11), name: varchar(100), content: text, map: text, phone: varchar(15), email: varchar(100), hours: varchar(100)
laravel_business icons_info	id_icon: int(11), title: varchar(100), content: text, icon: varchar(50), date: timestamp
laravel_business password_resets	email: varchar(255), token: varchar(255), created_at: timestamp
laravel_business success_projects	id_success: int(11), title: varchar(100), content: text, image: varchar(150), date: timestamp
laravel_business social_media	id_socialmedia: int(11), name: varchar(50), url: varchar(150), date: timestamp
laravel_business client	id_client: int(11), name: varchar(150), content: text, image: varchar(150), date: timestamp
laravel_business blog	id_blog: int(11), title: text, content: text, image: varchar(100), date: timestamp, important: int(1), visible: int(1)
laravel_business failed_jobs	id: bigint(20) unsigned, connection: text, queue: text, payload: longtext, exception: longtext, failed_at: timestamp
laravel_business migrations	id: int(10) unsigned, migration: varchar(255), batch: int(11)
laravel_business services	id_service: int(11), title: varchar(100), content: text, image: varchar(150), date: timestamp
laravel_business info_generic	id_info: int(11), name: varchar(150), content: text, date: timestamp
laravel_business company	id_company: int(11), name: text, content: text, image: varchar(100), date: timestamp
laravel_business users	id: bigint(20) unsigned, name: varchar(255), email: varchar(255), email_verified_at: timestamp, password: varchar(255), remember_token: varchar(100), created_at: timestamp, updated_at: timestamp

Figura 5: Taules de la base de dades

Com es pot veure a la figura anterior, no hi ha cap relació entre taules donat a que cada taula és única i no tenen cap relació entre elles, és a dir, entre elles contenen dades que no tenen res a veure amb la resta d'informació. Pel que fa a les claus primàries PK, són indicades amb una clau a cada primera columna.

Diagrama de navegació de les plantilles web

La plantilla de la pàgina web es distribueix en els següents apartats:

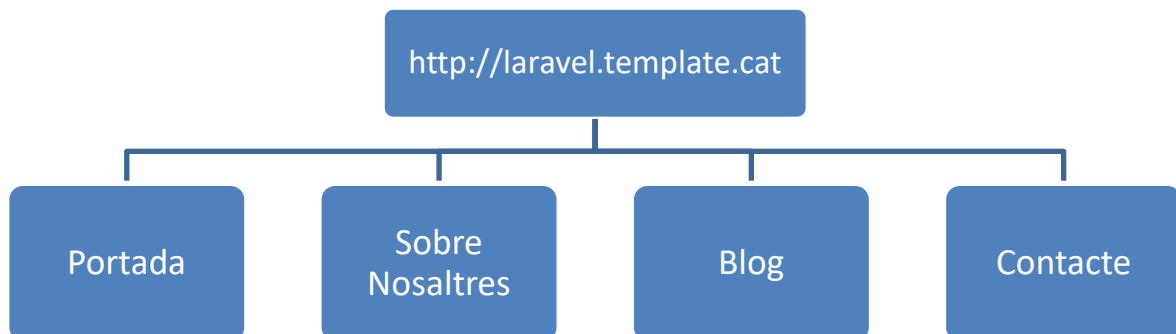


Figura 6: Diagrama de navegació de la plana web (Front-end).

La plantilla del panell d'administració es distribueix en els següents apartats:

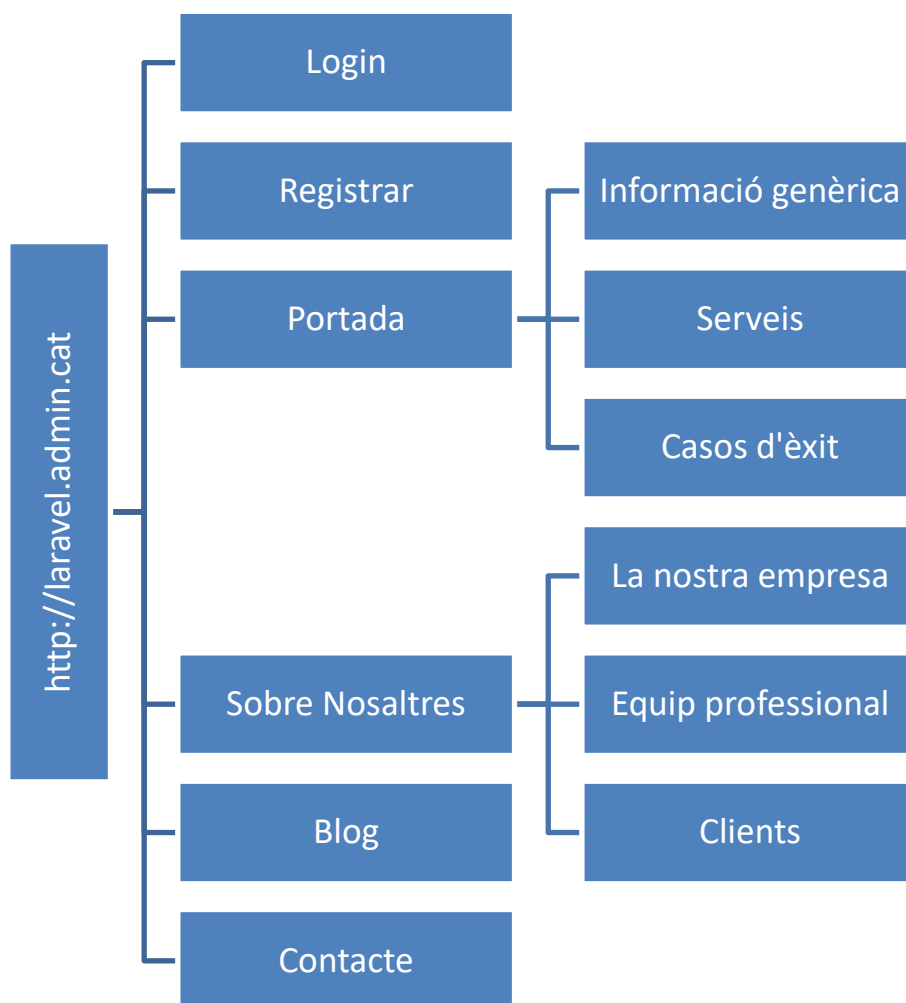


Figura 7: Diagrama de navegació del panell d'administració (BackOffice).

3. Disseny gràfic i interfícies

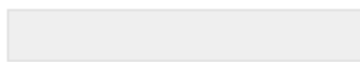
3.1 Estils

Definició de la línia gràfica del treball de la plantilla web (front-end):

- **Paleta de colors** formada principalment per:



Blanc: #fff



Gris clar: #efefef



Gris fosc: #343a40

- **Paleta tipogràfica**, grandària i estil de fonts:
 - Font de “Google Fonts” de la família “Montserrat”.
 - Grandària de font estàndard produïda pel framework Bootstrap.
- **Fons, icones, botons i altres elements gràfics:**
 - Icones introduïdes amb el paquet de recursos gratuïts Font Awesome versió 5.
 - Gràfics per defecte de Bootstrap 4.
 - Imatges sense copyright des de la web de Pexels.

3.2 Usabilitat i UX

El primer que s'ha de decidir a l'hora de dissenyar qualsevol pàgina web, és definir els apartats que tindrà, l'arbre de navegació i l'estil gràfic que tindrà. Donat que aquest projecte es basa en la funcionalitat de Docker i els seus contenidors que donaran servei a les plantilles web, aquestes han sigut dissenyades mitjançant altres plantilles simples predefinides pel propi Bootstrap:

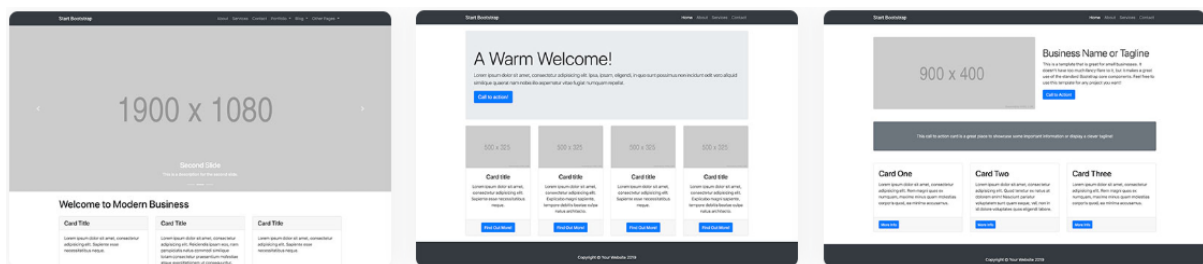


Figura 8: Plantilles bàsiques de Bootstrap

4. Llenguatges de programació

Donat que la finalitat d'aquest Treball Final de Màster és proveir d'una eina de suport al programador, s'han fet servir eines i programari gratuïtes i amb drets lliure d'utilització, modificació, fusió i distribució (licència MIT).

A continuació es detalla la informació sobre els recursos tecnològics utilitzats:

Software de desenvolupament

- **Visual Studio Code** i **Sublime Text 3**, per codificar les plantilles de les planes web.
- **VirtualBox**, programari per virtualitzar el Sistema Operatiu Ubuntu Server versió 18.04 LTS, amb suport de llarga durada.
- **Krita** i **Adobe Photoshop CC**, programa d'edició d'imatges.
- Navegadors **Google Chrome**, **Mozilla Firefox** i **Brave** on s'han realitzat les proves de funcionament de les planes web.
- **Putty**, programari gratuït d'accés a la consola del servidor virtual que dona millor comoditat a l'hora d'escriure les comandes, principalment per fer les proves amb els contenidors Docker.
- **Docker** i **Laradock**, programari basat en contenidors independents entre si per executar diverses aplicacions. Laradock incorpora una gran quantitat de contenidors preparades per ser instal·lades i utilitzades per fer funcionar qualsevol aplicació web, sobre tot Laravel.
- **GitHub**, lloc web de gestió de projectes i control de versions.
- **Trello**, lloc web amb taulers on portar a cap el control de les tasques d'un projecte.

Llenguatges de programació

- La programació de les plantilles web s'ha integrat amb **Bootstrap 4**, una llibreria multiplataforma de codi obert on utilitza els llenguatges **HTML5**, **CSS3** i **JavaScript**, que permet crear llocs web adaptades per a qualsevol dispositiu i capaç d'integrar-se amb Laravel.
- Pel desenvolupament de les aplicacions, s'ha fet servir el framework **Laravel versió 6**.
- Mitjançant paquets de Laravel, s'ha fet servir **React.js** pel Login del panell d'administració i el framework de JavaScript **jQuery** per l'intercanvi de dades client – servidor.
- **Bash**, per la generació de l'arxiu de comandes per Linux.
- La persistència de dades de la pàgina web ha estat desenvolupat amb **MySQL** (o MariaDB) que és la tecnologia que més es fa servir per planes web senzilles i és accessible des del contenidor "**PhpMyAdmin**" mitjançant un enllaç al navegador.

Justificació de la decisió

La decisió definitiva de fer servir Laravel i no un altre, recau en que és un framework de PHP que ha crescut molt i ha guanyat molta rellevància en els últims temps, deguda a la seva manera de programar modular orientat al patró "Model - Vista - Controlador", i que a diferència d'altres com Symfony o CodeIgniter on s'han quedat desfasades, aquest aporta més flexibilitat, robustesa, escalabilitat, senzillesa, rapidesa, seguretat i aporta moltes facilitats al programador, incloent la integració de codi extern creat per altres usuaris i que s'integren mitjançant plugins a la zona de treball.

Per altre costat, la decisió de fer servir Docker és que aporta gran avantatge sobre tot quan es tracta sobre servidors i aplicacions web. Actualment el que es fa servir és crear multitud de màquines virtuals úniques per publicar diferents aplicacions amb versions determinades, però amb Docker, només es cal crear una sola màquina virtual amb multitud de contenidors creats amb aplicacions de diferents versions, que no s'interferiran entre ells i que reaprofiten els recursos de la màquina i del Sistema Operatiu com a base comuna.

Capítol 4: Implementació

1. Requisits d'instal·lació

Per realitzar la instal·lació de l'eina de suport al programador, cal tindre la següent configuració:

Hardware

- El maquinari que necessita ha de ser una d'aquestes opcions: un servidor, un ordinador personal de torre o ordinador personal portàtil.

Software

- Per la instal·lació del software, és necessari tindre un servidor físic o virtual. En cas de ser un servidor virtual, es pot fer servir el programari gratuït VirtualBox d'Oracle.
- El Sistema Operatiu que s'ha d'instal·lar ha de ser preferentment amb Ubuntu Server versió 18.04 LTS.
- Dins de l'Ubuntu Server, tindre instal·lat openSSH per poder connectar-se amb el programari Filezilla Client o Putty.
- Per executar l'arxiu script de comandes `jhernandezch_script.sh`, es pot fer servir el programari Putty per una millor experiència amb la consola del sistema.
- Per la transferència de fitxers entre client i servidor, el programari Filezilla Client pot fer la connexió accedint al servidor mitjançant el port 22.

Formació i coneixements

- Coneixements bàsics amb les comandes i lleugeresa amb sistemes Linux.
- Coneixements acceptables de programació web.

2. Instruccions d'instal·lació

Els passos que s'han de seguir per fer la instal·lació de l'eina de suport al programador i que són els mateixos que s'han fet servir durant el desenvolupament, simplificadament són:

1. Instal·lació del programari VirtualBox.
2. Creant una nova màquina virtual, fer la instal·lació del Sistema Operatiu Ubuntu Server versió 18.04 LTS.
3. Dins d'Ubuntu Server, actualitzar el sistema i instal·lar el programari openSSH.
4. Mitjançant Filezilla, traspasar l'arxiu script `jhernandezch_script.sh` i executar-lo en bash.
5. Esperar a que faci tota la instal·lació automàtica de Docker, Laradock, les plantilles web i configuracions, fins que el servidor torni a l'espera de noves instruccions o comandes.
6. Modificar, si cal, l'arxiu `hosts` del sistema client per accedir a les URL que s'han definit a APACHE2 sobre l'arxiu `laravel_admin.conf` i `laravel_template.conf` situat dins del contenidor Docker Apache (i que el descarrega ja preconfigurat des del repositori GitHub), que són:
 - a. Font-end: <http://laravel.template.cat>
 - b. BackOffice: <http://laravel.admin.cat>
 - c. PhpMyAdmin (Base de dades): <http://laravel.template.cat:8080>
7. Des del client, obrir el navegador i introduir els enllaços anteriors.

Per veure la instal·lació més detallada, s'adjunta amb aquest document una **guia d'usuari** on s'explica en detall tot el procés d'instal·lació anomenat: **`jhernandezch_guia_usuario.pdf`**.

Capítol 5: Demostració

1. Instruccions d'ús

Per poder fer la instal·lació del present projecte, cal tindre com a mínim un servidor virtual en local, mitjançant per exemple amb el programari gratuït VirtualBox d'Oracle i, si es necessari, es pot seguir una instal·lació guiada amb l'ajuda de la guia d'usuari que s'adjunta amb aquest document: jhernandezch_guia_usuari.pdf. Amb aquest servidor i amb l'script que s'adjunta amb aquest document: jhernandezch_script.sh, cal executar-lo per comandes de la següent manera mitjançant bash:

```
sudo bash ./jhernandezch_script.sh
```

I ha continuació, esperar que faci tota la instal·lació pertinent fins que finalitzi el procés, que acabarà sortint la informació dels contenidors que s'han executat:

```
joan@tfm2019:/var/www/laradock$ sudo docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
b41dd74da786   laradock_apache2    "/opt/docker/bin/ent..." About a minute ago Up About a minute 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp
67d666eb5240   laradock_php-fpm    "docker-php-entrypoi..." About a minute ago Up About a minute 9000/tcp
f9b619723bf8   laradock_phpmyadmin "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:8080->80/tcp
182efba642ba   laradock_workspace  "/sbin/my_init"         About a minute ago Up About a minute 0.0.0.0:2222->22/tcp
40b8d55c3c6a   docker:dind         "dockerd-entrypoint..." About a minute ago Up About a minute 2375-2376/tcp
c80755a38b49   laradock_mysql      "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:3306->3306/tcp, 33060/tcp
```

Figura 9: Resultat final del procés d'instal·lació de l'script

Per a la demostració pública, s'han creat dos URL de forma temporal per poder veure que, mitjançant els contenidors Docker que s'han configurat, es mostren publicades les dues plantilles web:

- **Front-end:** <https://laravel.sabarca.cat:9443>
- **BackOffice:** <http://laravel.sabarca.cat:9444>

Pel que fa a la demostració privada amb el servidor virtual en local, les URL que corresponen són:

- **Front-end:** <http://laravel.template.cat>
- **BackOffice:** <http://laravel.admin.cat>
- **PhpMyAdmin (Base de dades):** <http://laravel.template.cat:8080>

Cal tenir en compte però que les URL correspon segons la IP de cada màquina virtual i la manera en que s'ha configurat el servidor APACHE i la modificació de l'arxiu "hosts" de la màquina client, tal com s'indica al següent punt de tests.

2. Tests

Les proves que s'han realitzat en aquest projecte s'han basat sobre tot en l'execució dels contenidors Dockers lligat amb les pàgines web, on sense una bona configuració tant del servidor web APACHE, com de la base de dades MySQL i la tecnologia PHP, les pàgines no haurien funcionat mai.

Així doncs, per realitzar les proves, es va configurar en local un nou servidor virtual mitjançant el programa VirtualBox d'Oracle, amb un sistema operatiu Linux Ubuntu Server en versió 18.04 LTS. Un cop instal·lat i actualitzat el servidor, s'ha executat l'script de tal manera que fes tot el procés des d'inici fins a la publicació de les pàgines.

```
Creating network "laradock_frontend" with driver "bridge"
Creating network "laradock_backend" with driver "bridge"
Creating network "laradock_default" with the default driver
Creating laradock_mysql_1 ... done
Creating laradock_docker-in-docker_1 ... done
Creating laradock_workspace_1 ... done
Creating laradock_phpmyadmin_1 ... done
Creating laradock_php-fpm_1 ... done
Creating laradock_apache2_1 ... done
```

Figura 10: Creació dels contenidors Docker

```
joan@tfm2019:/var/www/laradock$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
b41dd74da786      laradock_apache2  "/opt/docker/bin/ent...  About a minute ago  Up About a minute
67d666eb5240      laradock_php-fpm  "docker-php-entrypoi...  About a minute ago  Up About a minute
f9b619723bf8      laradock_phpmyadmin  "/docker-entrypoint...  About a minute ago  Up About a minute
102efba642ba      laradock_workspace  "/sbin/my_init"      About a minute ago  Up About a minute
40b8d55c3c6a      docker:dind        "dockerd-entrypoint...  About a minute ago  Up About a minute
c80755a38b49      laradock_mysql    "docker-entrypoint.s...  About a minute ago  Up About a minute
```

Figura 11: Contenedors Docker funcionant

Per accedir a les pàgines de manera local, des de l'ordinador client cal configurar l'arxiu "hosts" indicant la IP del servidor i el nom que s'ha associat amb APACHE. En el moment de les proves, s'ha fet l'accés des d'un sistema operatiu Windows 10 on l'arxiu "hosts" s'ubica a: "C:\Windows\System32\drivers\etc". El contingut d'aquest arxiu en el moment de les proves han sigut:

```
192.168.1.14    laravel.template.cat
192.168.1.14    laravel.admin.cat
```

Figura 12: Contingut de l'arxiu hosts

Amb el següent resultat:

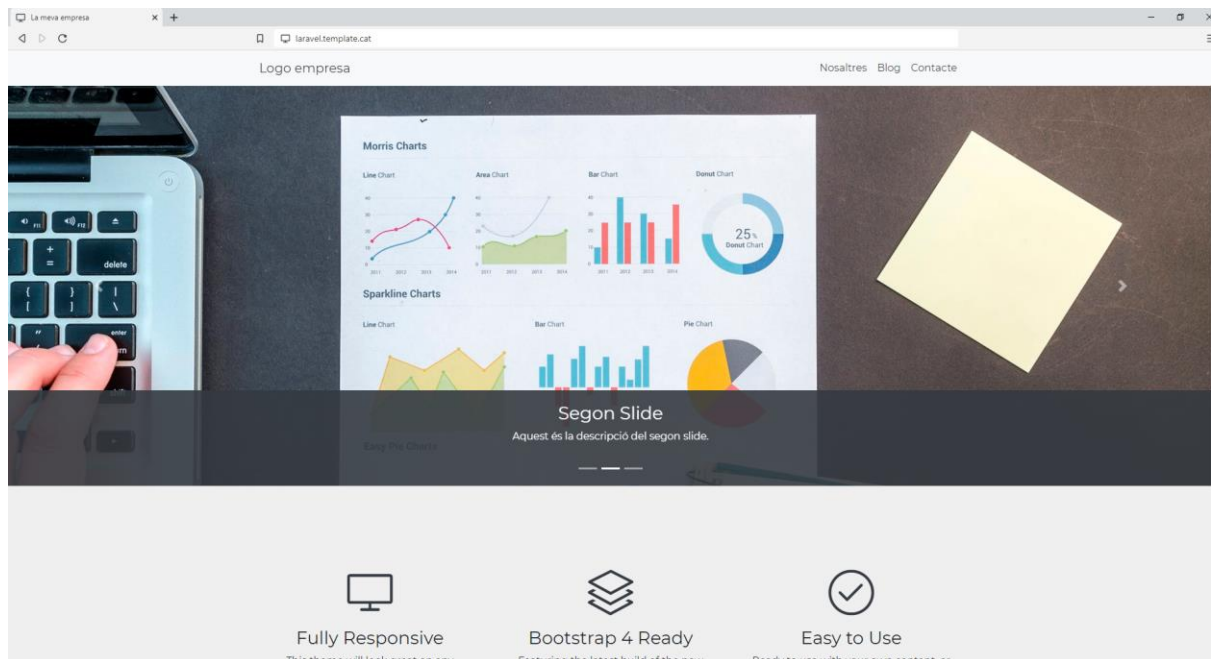


Figura 13: Captura del contingut del front-end

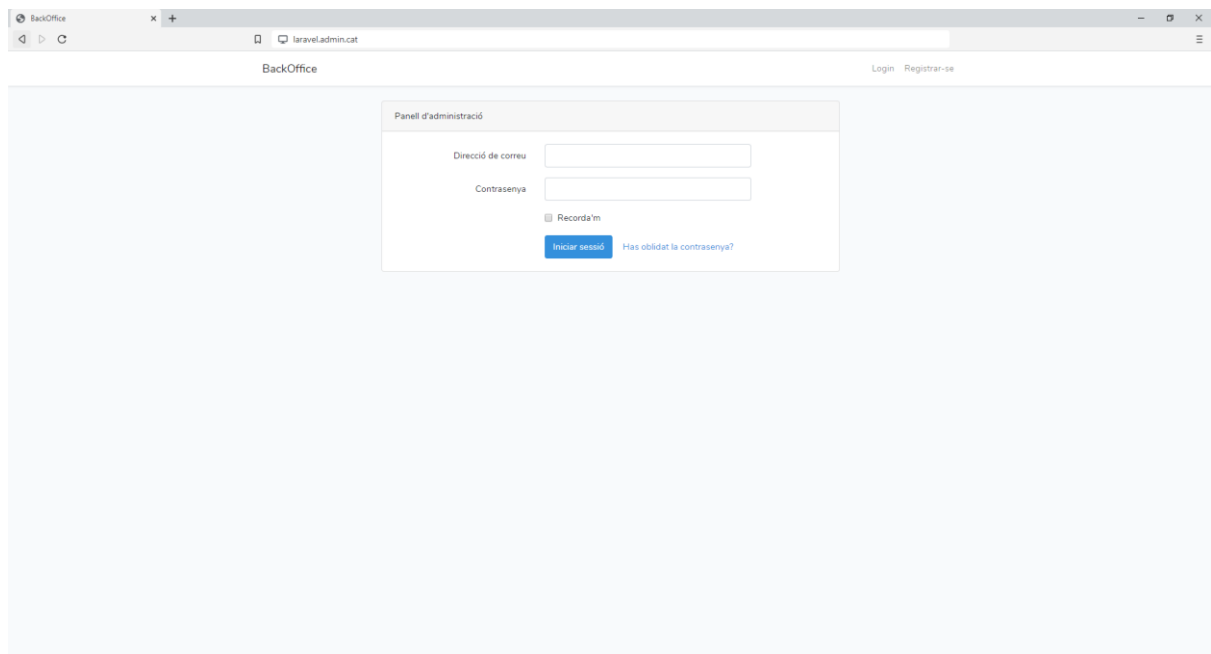


Figura 14: Captura del contingut del backoffice

4. Exemples d'ús del producte

Per obtenir tot el paquet de recursos instal·lats i configurats, l'arxiu adjunt jhernandezch_script.sh conté totes les comandes necessàries següents:

```
#!/bin/bash
echo "---> 1 - Instal·lació Laravel Admin"
sudo git clone https://github.com/jrhernandezch/laravel_admin.git /var/www/laravel_admin
sudo chmod 777 -R /var/www/laravel_admin/storage
sudo chmod 777 -R /var/www/laravel_admin/bootstrap/cache
echo "Completada"

echo "---> 2 - Instal·lació Laravel Template"
sudo git clone https://github.com/jrhernandezch/laravel_template.git /var/www/laravel_template
sudo chmod 777 -R /var/www/laravel_template/storage
sudo chmod 777 -R /var/www/laravel_template/bootstrap/cache
sudo chmod 777 -Rf /var/www/laravel_template/public/img
echo "Completada"

echo "---> 3 - Instal·lació Docker"
sudo apt-get update
sudo apt-get remove docker docker-engine docker.io
sudo apt install docker.io
sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
sudo systemctl start docker
sudo systemctl enable docker
echo "Completada"

echo "---> 4 - Instal·lació Laradock a Laravel template"
sudo git clone https://github.com/Laradock/laradock.git /var/www/laradock
echo "Completada"

echo "---> 5 - Instal·lació recursos laradock"
sudo cp /var/www/laravel_template/laradock_resources/.env /var/www/laradock/.env
sudo cp /var/www/laravel_template/laradock_resources/my.cnf /var/www/laradock/mysql/my.cnf
sudo cp /var/www/laravel_template/laradock_resources/createdb.sql /var/www/laradock/mysql/docker-entrypoint-initdb.d/createdb.sql
sudo cp /var/www/laravel_template/laradock_resources/laravel_admin.conf /var/www/laradock/apache2/sites/laravel_admin.conf
sudo cp /var/www/laravel_template/laradock_resources/laravel_template.conf /var/www/laradock/apache2/sites/laravel_template.conf
echo "Completada"

echo "---> Instal·lació finalitzada. Inicialitzant contenidors Docker"
cd /var/www/laradock
sudo docker-compose up -d apache2
sudo docker-compose up -d mysql
sudo docker-compose up -d phpmyadmin
sudo docker-compose up -d workspace
sudo docker ps
```

Figura 15: Comandes de servidor en bash

I per executar aquest script, només cal entrar al servidor on es vol fer la instal·lació i executar la comanda següent:

```
sudo bash ./jhernandezch_script.sh
```

Per tant, el servidor ha de tindre la següent estructura definida:

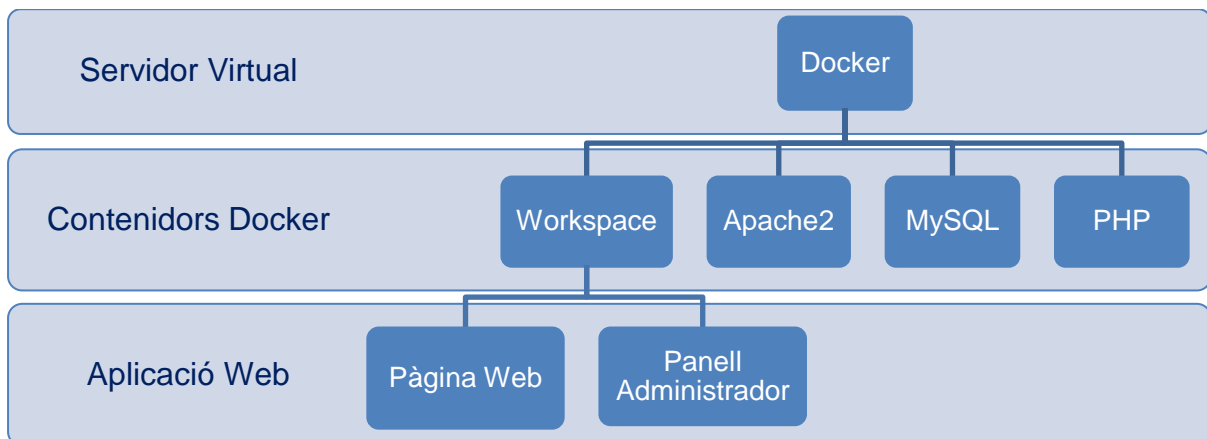


Figura 16: Estructura del servidor virtual

Capítol 6: Conclusions i línies de futur

1. Conclusions

Realitzar tot el desenvolupament d'una estructura abstracta que d'entrada mai havia fet servir, a nivell personal i professional ha sigut tot un repte i donar-li forma i acabant amb unes pàgines web funcionals i suportats per contenidors Docker, és una gran satisfacció. A més, planificar i desenvolupar tot un projecte des de zero ajuda molt a pensar i prendre decisions que si es fessin en el moment ja que el temps de dedicació seria major del que ha de ser i d'aquesta manera, ja es pot tindre una pauta de seguiment.

La lliçó més important d'aquest projecte no és aquella tecnologia que s'aprèn sinó que és la gestió i inversió del temps: És molt difícil calcular el temps de duració d'un projecte d'inici a fi i amb les seves fites. En cas de tindre experiència en la temàtica, pot ser que calcular el temps sigui més fàcil però de la manera que s'ha de calcular sobre temes que mai s'han tractat, resulta ser més difícil i impredecible. Però, sense dubte l'esforç ha valgut la pena.

Gràcies a les prioritats que es van establir a l'inici del projecte s'han assolit tots els objectius que es van proposar: Fer una instal·lació automàtica d'un desplegament Docker amb plantilles web i publicar-les. A més, haver après Docker va ser molt agradable i més quan vaig conèixer eines com "Laradock" que facilita encara més la utilització de contenidors dedicats al meu propòsit. Gràcies a tot això, a nivell personal puc agafar les plantilles per realitzar altres projectes amb un gran estalvi de temps.

A nivell de planificació, es va tindre que retocar la part d'usabilitat de les tecnologies de front-end, en que la idea inicial era aprendre i fer servir frameworks de JavaScript com React, Angular i Vue però que per manca de temps i que la dedicació en aquest sentit havia de ser més curt i simple i dedicar més recursos a Docker, es va tindre que fer un canvi a Bootstrap amb jQuery i Ajax i deixar que en un futur pròxim, la part visual sigui amb aquest frameworks.

Tot i que s'acaba aquest projecte, la intenció al llarg del temps és anar millorant tot el contingut de l'eina de suport fins arribar a un punt on encara serà més potent i amb més estalvi de temps que suposarà per a tot programador web.

2. Línies de futur

Es preveu que els recursos que s'han desenvolupat durant el present Treball Final de Màster és en una fase inicial amb la idea de fer ampliacions i millores de forma permanent, ja que no hi ha cap tecnologia web que no tingui actualitzacions en el temps.

Un cop finalitzada la data final d'aquest TFM, les millores que vindran a curt termini durant els pròxims mesos serà l'ampliació de funcionalitats de les dues pàgines webs: front-end i panell d'administració. Mentre que a mig termini les millores seran de les configuracions de Docker amb l'eina Laradock, especialment amb la integració de Laravel.

Bibliografia

- [1] Docker, Muntar, compartir i córrer aplicacions moderns en qualsevol lloc <https://www.docker.com>, consultat 11/10/2019.
- [2] HTML5, HyperText Markup Language, versió 5, Documentació per desenvolupadors Firefox <https://developer.mozilla.org/es/docs/HTML/HTML5>, consultat 11/10/2019.
- [3] CSS3, Full d'estils en Cascada, documentació desenvolupadors Firefox <https://developer.mozilla.org/es/docs/Archive/CSS3>, consultat 11/10/2019.
- [4] JavaScript, MDN Web doc <https://developer.mozilla.org/es/docs/Web/JavaScript>, consultat 11/10/2019.
- [5] Bootstrap, Eina de desenvolupament web <https://getbootstrap.com>, consultat 11/10/2019.
- [6] Material Design, Clàssics principis del bon disseny amb tecnologia de la innovació i ciència <https://material.io/design>, consultat 11/10/2019.
- [7] Materialize, Framework per front-end responsive basat en Material Design <https://materializecss.com>, consultat 11/10/2019.
- [8] Angular, Framework JavaScript per mòbil i ordinador <https://angular.io>, consultat 11/10/2019.
- [9] React, Biblioteca JavaScript per interfícies d'usuari <https://es.reactjs.org>, consultat 11/10/2019.
- [10] VUE, Framework de JavaScript progressiu <https://vuejs.org>, consultat 11/10/2019.
- [11] PHP, Manual de PHP <https://www.php.net/manual/es/intro-what-is.php>, consultat 11/10/2019.
- [12] Laravel, Framework de PHP <https://laravel.com>, consultat 11/10/2019.
- [13] Django, The web framework for perfectionists with deadlines <https://www.djangoproject.com>, consultat 11/10/2019.
- [14] MySQL, Open Source database <https://www.mysql.com>, consultat 11/10/2019.
- [15] MariaDB, The MariaDB Foundation database <https://mariadb.org>, consultat 11/10/2019.
- [16] MongoDB, Base de dades per aplicacions modernes <https://www.mongodb.com/es>, consultat 11/10/2019.
- [17] PostgreSQL, Advanced Open Source Relational Database <https://www.postgresql.org>, consultat 11/10/2019.
- [18] Contenedors Docker, Funcionament del sistema entre Sistema Operatiu i Docker [https://es.wikipedia.org/wiki/Docker_\(software\)](https://es.wikipedia.org/wiki/Docker_(software)), consultat 11/10/2019.
- [19] Apache2, Servidor HTTP <https://httpd.apache.org>, consultat 11/10/2019.
- [20] JQuery, Llibreria basat en JavaScript, <https://jquery.com>, consultat 11/10/2019.
- [21] Start Bootstrap, Ajuda amb projectes Bootstrap <https://startbootstrap.com>, consultat 23/10/2019.
- [22] AdminLTE, Bootstrap Admin Dashboard Template <https://adminlte.io>, consultat 5/11/2019.
- [23] Scrum, Proyectos ágiles <https://proyectosagiles.org/que-es-scrum>, consultat 12/10/2019.
- [24] Trello, Projectes organitzats <https://trello.com>, consultat 13/10/2019.
- [25] Kanban, Kanban tool <https://kanbantool.com/es/metodologia-kanban>, consultat 13/10/2019.
- [26] Laradock, Paquet Docker per Laravel <https://laradock.io>, consultat 9/12/2019.
- [27] VirtualBox, Software de virtualització gratuïta <https://www.virtualbox.org>, consultat 11/10/2019.
- [28] Ubuntu Server, <https://ubuntu.com/download/server>, consultat 10/10/2019.

- [29] Visual Studio Code, Code editin. Refined. <https://code.visualstudio.com/>, consultat 11/10/2019.
- [30] SublimeText3, A sophisticated text editor for code <https://www.sublimetext.com>, consultat 11/10/2019.
- [31] Font Awesome 5, The web's most popular icon set and Toolkit <https://fontawesome.com>, consultat 11/10/2019.
- [32] Pexels, The best free images <https://www.pexels.com>, consultat 11/10/2019.
- [33] Bootstrap, Amb frameworks JavaScript, <https://ng-bootstrap.github.io>, consultat 11/10/2019.
- [34] BootstrapMD, Material Design per Bootstrap <https://mdbbootstrap.com>, consultat 11/10/2019.
- [35] Material Design amb Angular, Material Design components for Angular <https://material.angular.io>, consultat 11/10/2019.
- [36] VueJS,
- [37] Nginx, Servidor web/proxy <https://www.nginx.com>, consultat 11/10/2019.
- [38] Tomcat, Servidor d'aplicacions Java Servlets <http://tomcat.apache.org>, consultat 11/10/2019.
- [39] Glassfish, Servidor d'aplicacions Java Enterprise Edition (JEE) <https://javaee.github.io/glassfish>, consultat 11/10/2019.
- [40] Python, Python Software Foundation <https://www.python.org>, consultat 11/10/2019.
- [41] Java, Java Platform <https://www.oracle.com/java/technologies/java-ee-glance.html>, consultat 11/10/2019.
- [42] SQL, Oracle Database <https://www.oracle.com/es/database>, consultat 13/10/2019.
- [43] Citrix, Hipervisor de màquines virtuals (VDI) <https://www.citrix.es/digital-workspace/virtualization-vdi.html>, consultat 11/10/2019.
- [44] VMWare, Software de virtualització privada <https://www.vmware.com>, consultat 11/10/2019.

Annexos

Annex A: Glossari

- **TFM:** Treball Final de Màster.
- **Front-end:** Desenvolupament gràfic en el costat del client / navegador.
- **Back-end:** Desenvolupament en el costat del servidor.
- **BackOffice:** Pàgina web panell d'administrador.
- **HTML:** Llenguatge de marques pel desenvolupament web.
- **CSS:** Cascading StyleSheets, fulls d'estils que complementa l'HTML.
- **SQL:** Llenguatge de consulta estructurada de bases de dades.
- **Script bash:** Arxiu que s'executa amb un codi font programat en llenguatge bash.
- **Framework:** Esquema, esquelet o patró preparat prèviament per al desenvolupament i/o la implementació d'una aplicació.
- **URL:** Enllaç a una pàgina web des d'un navegador.

Annex B: Lliurables del projecte

Arxiu comprimit ZIP - jhernandezch_TFM.zip:

- **jhernandezch_memoria.pdf:** Memòria del projecte.
- **jhernandezch_guia_instalacio.pdf:** Guia d'usuari d'instal·lació i proves del projecte.
- **jhernandezch_presentacio_academica.mp4:** Vídeo de la presentació del projecte.
- **jhernandezch_presentacio_publica.ppsx:** PowerPoint de presentació del projecte.
- **jhernandezch_presentacio_publica_audio.pptx i sense àudio.**
- **jhernandezch_script.sh:** Per executar el procés d'instal·lació de l'eina d'ajuda al programador, s'adjunta un arxiu script amb totes les comandes necessàries.
- **laravel_admin.zip:** Plantilla del panell d'administrador.
- **laravel_template.zip:** Plantilla de la plana web (Front-end).
- **Laradock_resources:** Directori amb les configuracions específiques pels contenidors Docker:
 - **jhernandezch_base_de_dades.sql:** Arxiu de base de dades amb taules i registres.
 - **.env:** Arxiu de configuració de Laradock sobre els contenidors Docker.
 - **createdb.sql:** Arxiu de configuració de Laradock per a la creació de la base de dades sobre el contenidor de MySQL.
 - **my.cnf:** Arxiu de configuració de Laradock per a la creació d'usuari "root" al contenidor de MySQL.
 - **laravel_admin.conf:** Arxiu de configuració de Laradock per l'accés a la plana web d'administració per al contenidor Apache2.
 - **laravel_template.conf:** Arxiu de configuració de Laradock per l'accés a la pàgina web per al contenidor Apache2.

Annex C: Captures de pantalla

Captura de l'eina gratuïta Trello: Eina que s'ha fet servir per portar el control del desenvolupament del projecte:

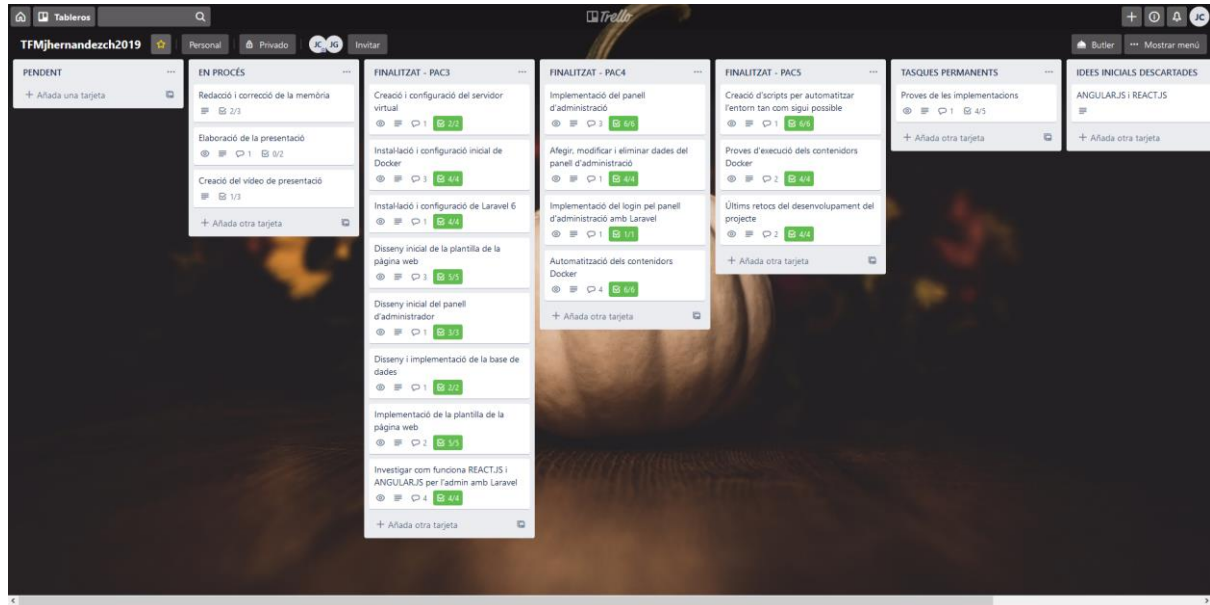


Figura 17: Trello, eina de control del desenvolupament

Captures de la simple plantilla creada pel front-end:

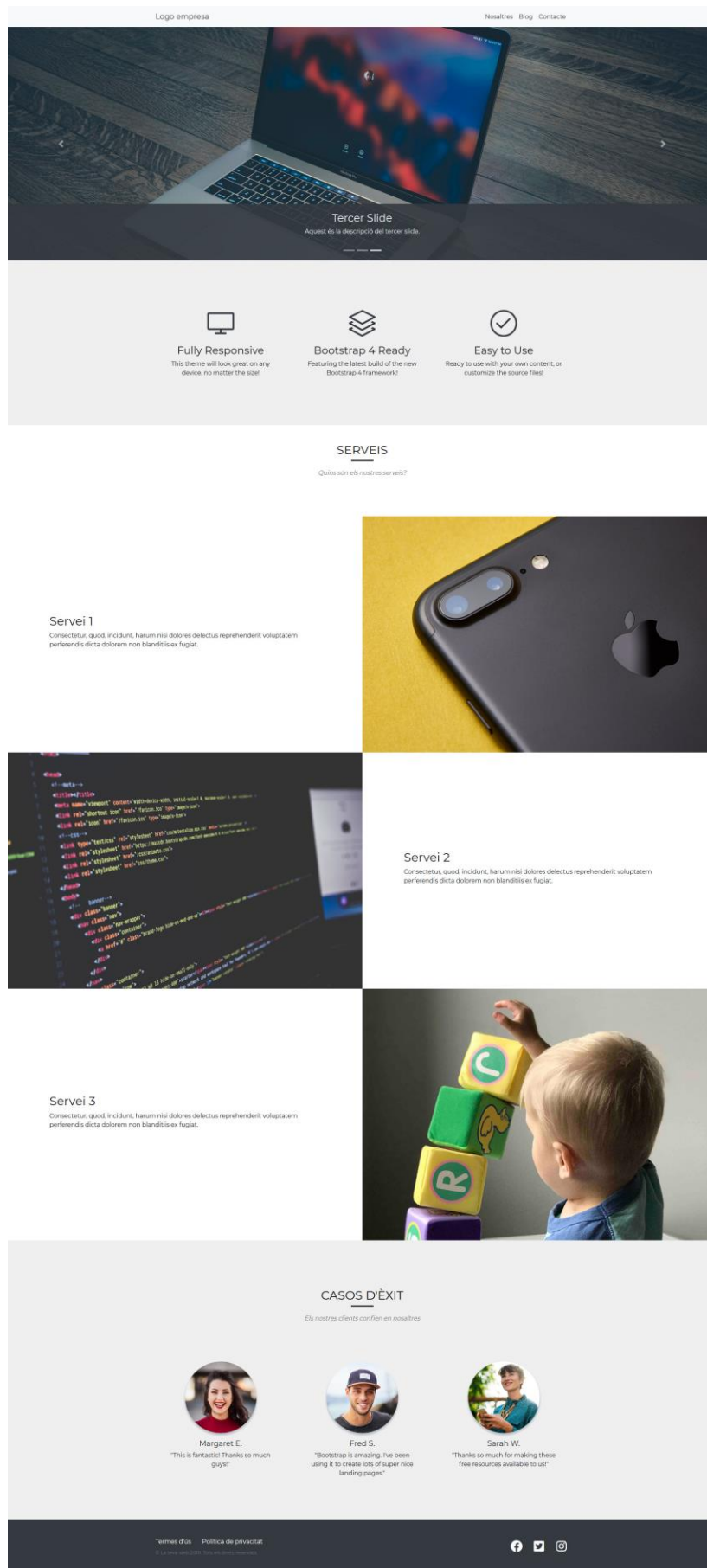


Figura 18: Front-end – Pàgina de portada

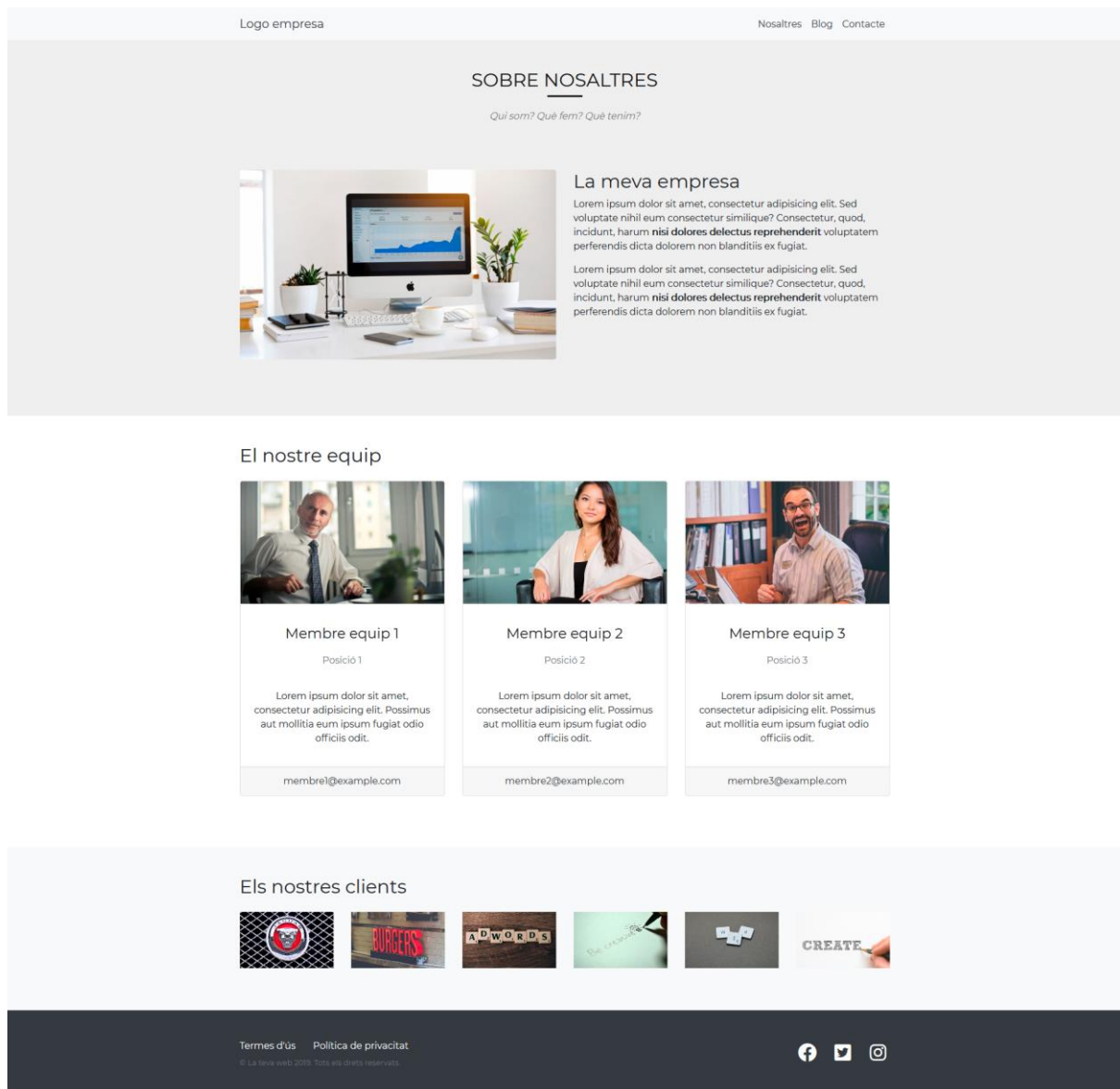


Figura 19: Front-end – Pàgina sobre nosaltres

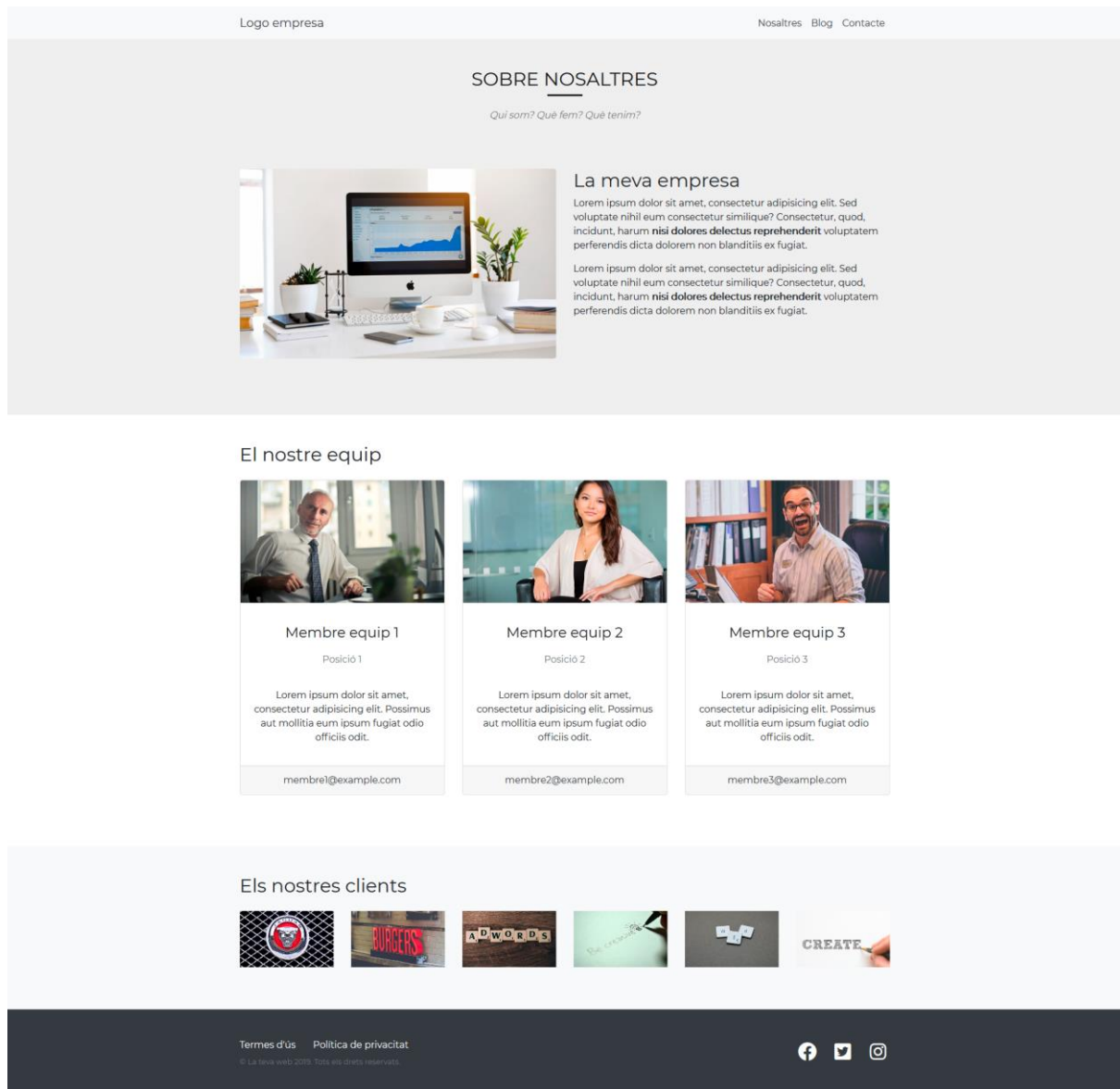


Figura 20: Front-end – Pàgina de blog

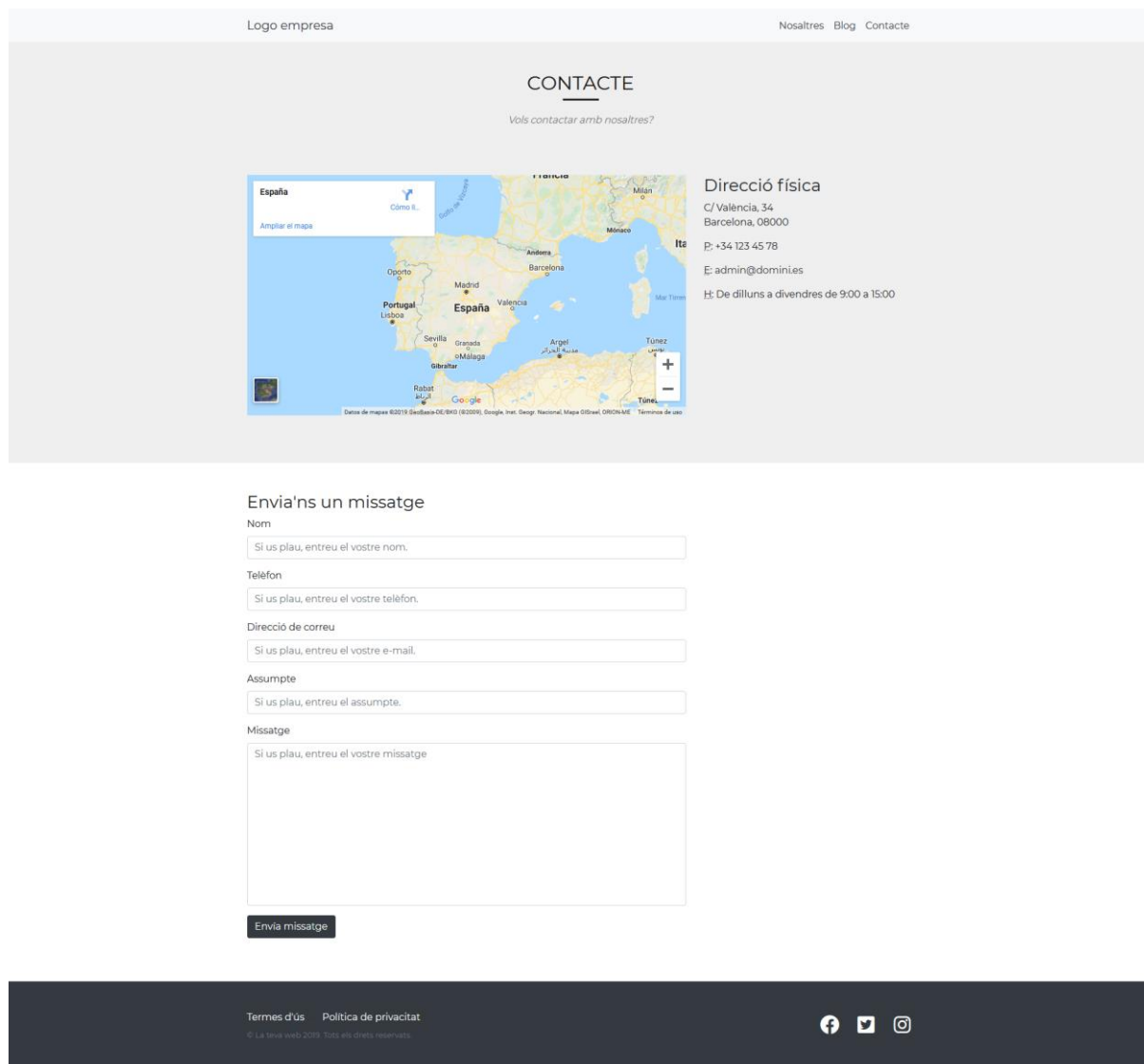


Figura 21: Front-end – Pàgina de contacte

Captures del panell d'administració, d'entre totes:

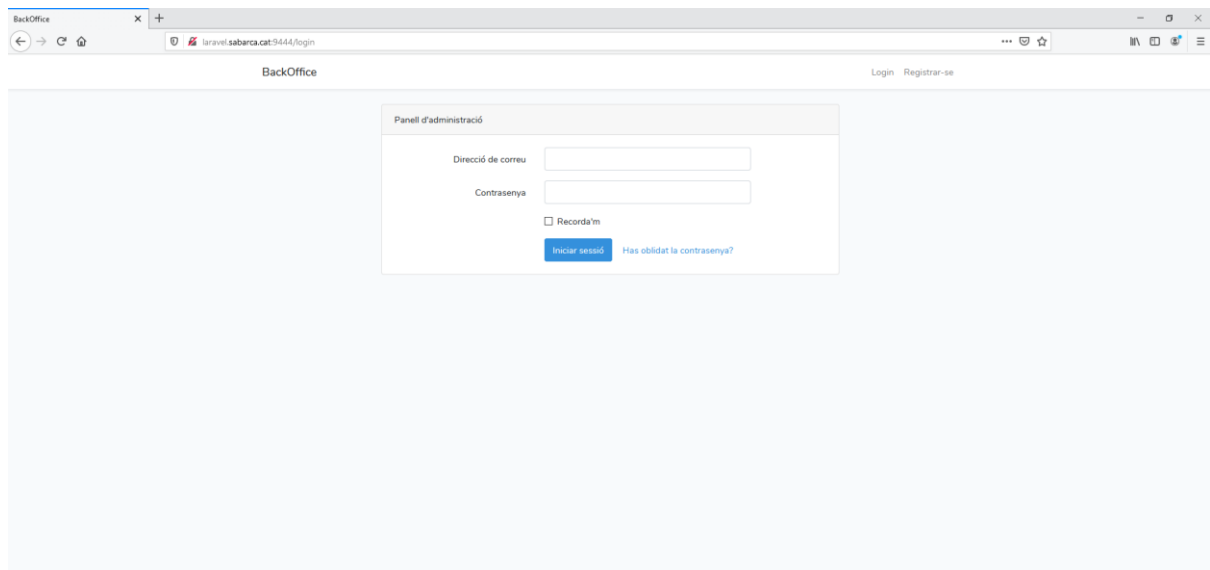


Figura 22: BackOffice – Pàgina de Login

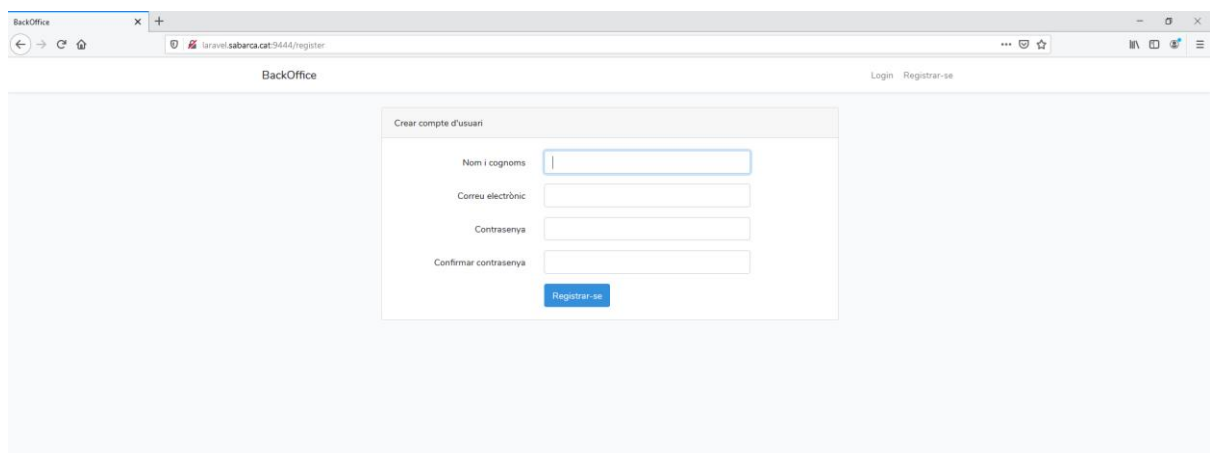


Figura 23: BackOffice – Pàgina de registre d'usuari

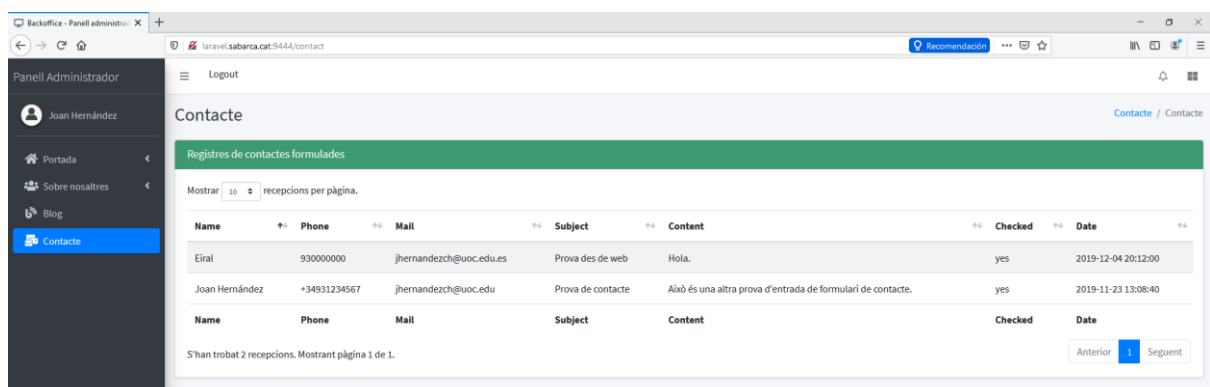


Figura 24: BackOffice – Pàgina de contactes

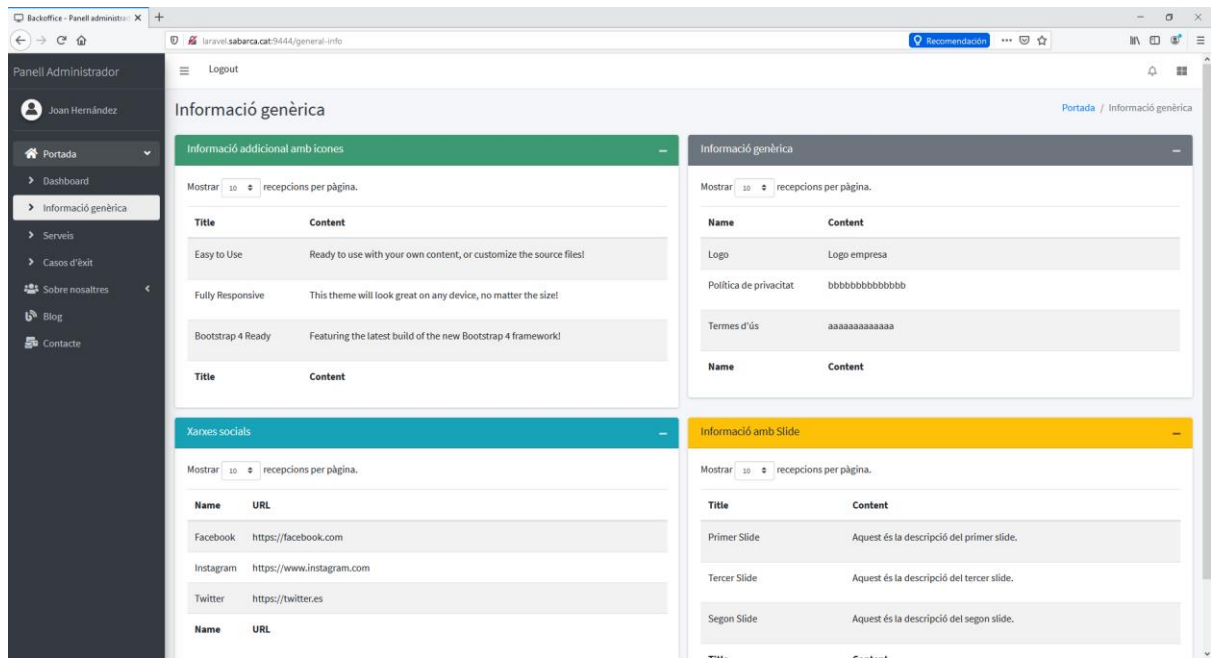


Figura 25: BackOffice – Pàgina d'informació genèrica

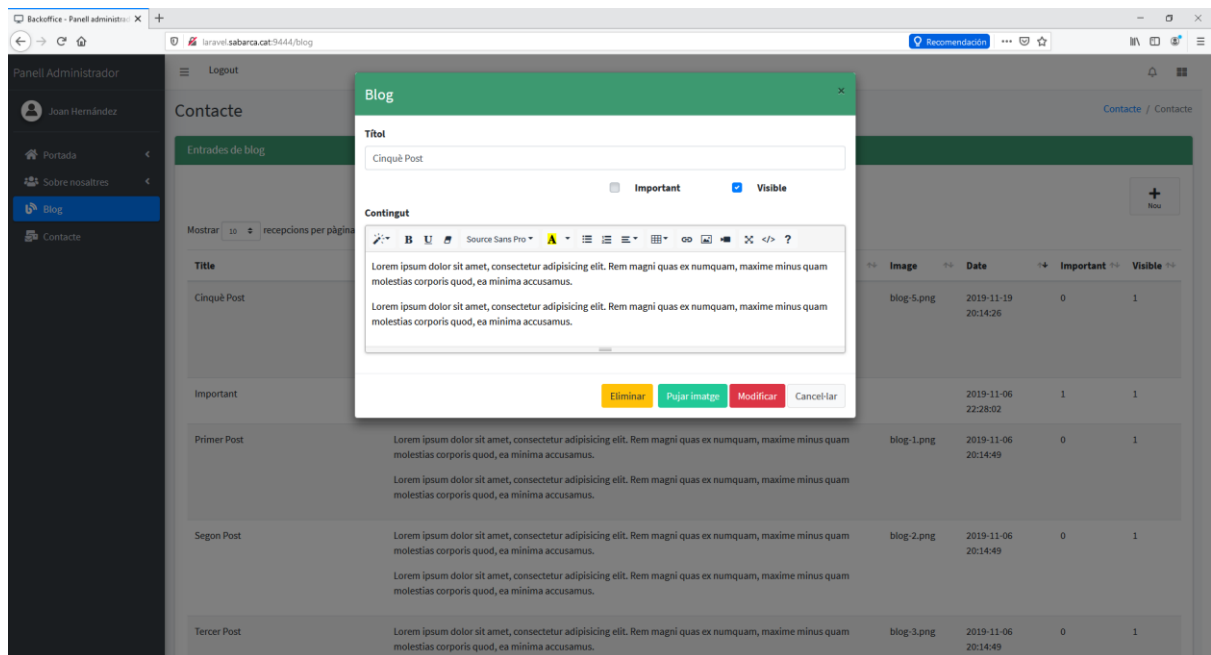


Figura 26: BackOffice – Pàgina de blog: Modificar contingut

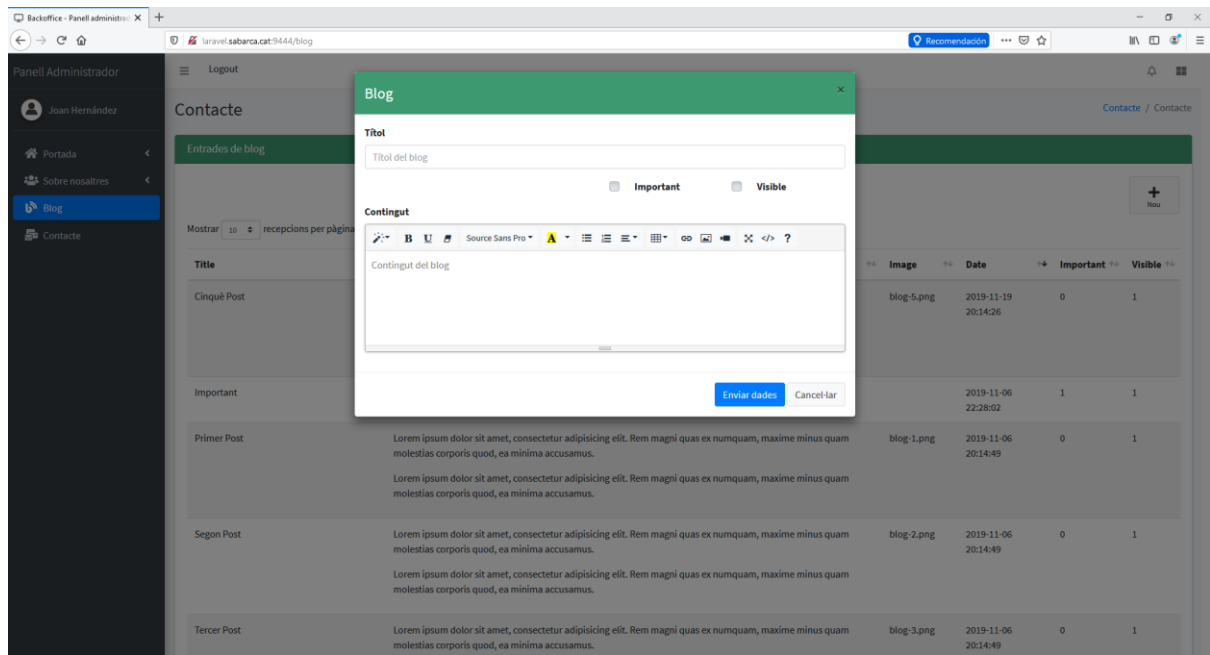


Figura 27: BackOffice – Pàgina de blog: Nou contingut