

Buidy: Comparador de components de hardware

Grau d'Enginyeria Informàtica

Desenvolupament Web

Autor: Joan Pedreño Marine

Consultor: Gregorio Robles Martínez

Professor responsable: Santi Caballé Llobet

12 de Juny de 2019



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

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Buidy: Comparador de components de hardware</i>
Nom de l'autor:	<i>Joan Pedreño Marine</i>
Nom del consultor/a:	<i>Gregorio Robles Martínez</i>
Nom del PRA:	<i>Santi Caballé Llobet</i>
Data de lliurament (mm/aaaa):	<i>06/2019</i>
Titulació o programa:	<i>Grau d'enginyeria informàtica</i>
Àrea del Treball Final:	<i>Desenvolupament web</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>WebApp, Comparador, Scraper</i>

Resum del Treball (màxim 250 paraules):

Aquest projecte neix de la voluntat de donar visibilitat als petits i mitjans comerços en lineal per tal de millorar la seva quota de mercat i, d'aquesta manera, evitar o minimitzar el inexorable creixements de les grans empreses que aspiren a monopolitzar el mercat.

Amb aquesta fita en ment, s'ha desenvolupat una aplicació web mitjançant un *stack* tecnològic modern conegut com MEAN (MongoDB, Express.js, Angular i Node.js), per ajudar als consumidors a comparar productes entre diferents tendes i al llarg del temps.

Per fer possible aquest objectiu, seguir la planificació ha sigut un punt fonamental degut a la fragmentació de les tasques en cadascun dels àmbits, així com també l'ús d'una metodologia àgil basada en cicles, que ha permès tenir control de les entregues i realitzar els canvis en les tasques necessaris abans de que fossin incorregibles de cara a l'entrega final.

Tenint present el pla de treball estipulat inicialment, tot i sabent que l'aplicació encara te punts per ampliar i millorar per arribar a ser un producte complet i llest per publicar, el resultat ha complert amb les expectatives i ha sigut satisfactori.

Abstract (in English, 250 words or less):

This project is born on the hope to provide visibility to online small businesses so they can increase their market share, to avoid or minimize the inexorable growth of large companies that aim to monopolize the market.

With this objective in mind, a web application with a modern technological stack known as MEAN (MongoDB, Express.js, Angular and Node.js) has been developed to help consumers by comparing products between different retailers and over time.

In order to achieve this goal, following the planning has been a fundamental point due to the fragmentation of tasks in each one of the areas, as well as the use of a sprint based agile methodology, which has provided control over deliveries and do the necessary changes to tasks before they were incorrigible keeping in mind the final delivery.

Bearing in mind the initially stipulated work plan, although knowing that the application still needs to expand and improve some functionalities to become a complete product and ready to publish, the result has met the expectations and has been satisfactory.

Índex

1. Introducció	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball.....	2
1.3 Enfocament i mètode seguit	3
1.4 Planificació del Treball	5
1.5 Breu sumari de productes obtinguts	8
1.6 Breu descripció dels altres capítols de la memòria.....	9
2. Arquitectura del sistema.....	10
2.1 Arquitectura general	10
2.2 Arquitectura del client web	12
2.2 Arquitectura del servidor	14
2.3 Esquema de la base de dades	16
2.4 Scraper	18
3. Disseny	21
3.1 Guia d'estils	21
3.2 Navegació	26
4. Desenvolupament	27
4.1 Llenguatges de programació	27
4.2 Eines de desenvolupament	28
4.3 Llibreries i <i>Frameworks</i>	29
5. Resultats Obtinguts.....	30
5.1 Pantalles i funcionalitats.....	30
5.2 Rendiment.....	36
6. Conclusions	38
7. Glossari.....	40
8. Bibliografia	41
9. Annexos	42
Annex 1: Portal i instal·lació local.....	42
Annex 2: Estructura del projecte	44

Llista de figures

Figura 1: Cicle desenvolupament àgil.....	4
Figura 2: Diagrama de Gantt 1	7
Figura 3: Diagrama de Gantt 2	7
Figura 4: Arquitectura general	11
Figura 5: Patró MVVM	12
Figura 6: Arquitectura de la capa de presentació	13
Figura 7: Arquitectura de la capa de negoci	15
Figura 8: Diagrama de classes	17
Figura 9: Scraping.....	18
Figura 10: Diagrama de flux.....	20
Figura 11: Paleta de colors	22
Figura 12: Logotip	22
Figura 13: Tipografia.....	23
Figura 14: Component barra superior.....	23
Figura 15: Component pestanyes.....	24
Figura 16: Component panell desplegable	24
Figura 17: Component taula	24
Figura 18: Component diàleg.....	25
Figura 19: Component notificació	25
Figura 20: Component formulari	25
Figura 21: Navegació.....	26
Figura 22: Landing principal.....	30
Figura 23: Diàleg registre.....	31
Figura 24: Menú components	31
Figura 25: Taula de productes	32
Figura 26: Detall de producte.....	33
Figura 27: Nou component	34
Figura 28: Zona usuari.....	35
Figura 29: Builds finalitzades	35
Figura 30: Vista builds	36
Figura 31: Prova rendiment Lighthouse.....	36
Figura 32: Prova rendiment Webpagetest	37
Figura 33: Prova rendiment GTmetrix.....	37

Llista de Taules

Taula 1: Cost serveis	5
Taula 2: Cost recursos	5
Taula 3: Planificació	6

1. Introducció

1.1 Context i justificació del Treball

Mica en mica i amb el pas dels anys la globalització, juntament amb les noves tecnologies, estan dotant de gran poder i influència a les grans empreses com Google o Amazon.

Aquestes, per tal de perpetrar la seva expansió, estan monopolitzant molts sectors com els serveis i productes d'oci entre d'altres fins al punt de tenir prou influència per poder generar pressió als poders legislatius dels governs.

Es per això que cada vegada es més difícil per les petites i mitjanes empreses poder competir i generar beneficis, ja que no disposen dels recursos per fer front a les grans campanyes publicitàries.

Per tal de prendre part en aquesta "lluita" i ajudar a donar visibilitat als petits i mitjans comerços de productes electrònics, vull desenvolupar una plataforma web gratuïta que permeti poder comparar diferents productes en l'àmbit de la informàtica de forma senzilla, perquè els consumidors puguin tenir una més objectiva i també garantir una competitivitat per tal d'assegurar un preu just.

1.2 Objectius del Treball

Podem dividir els objectius del treball en els següents grups:

Objectius generals

- L'objectiu d'aquest projecte es elaborar una aplicació web per comparar preus entre components de *hardware* mitjançant tecnologies web MEAN (MongoDB, Express.js, Angular i Node.js).

Objectius principals

- Dissenyar i implementar un portal web responsiu amb dispositius mòbils i que permeti interactuar amb els serveis que ofereix.
- Desenvolupar un servidor web que serveixi informació al client web.
- Idear un sistema de "*scraping*" per centralitzar la informació obtinguda en diverses webs per emmagatzemar-la al nostre servidor.
- Provar que amb l'aprenentatge adquirit, juntament amb la capacitat de autogestió, s'ha pogut elaborar un projecte o prototip amb prou qualitat per esdevenir un producte real.

Objectius secundaris

- Fer us d'un disseny modern, senzill de fer servir per l'usuari i mantenir coherència amb les interaccions com dicten les bones practiques de *User Experience (UX)*.
- Garantir un bon rendiment en l'aplicació i l'optimització de recursos, garantint que en cap cas la carrega superi els 5 segons amb un volum inferior als 100 usuaris connectats simultàniament.

- Elaborar un codi net, que prescindeixi de redundàncies i estigui correctament modular i cohesionat.
- Gestionar diferents permisos per a usuaris que han accedit al sistema dels convidats.
- Garantir que les operacions d'*scraping* no seran nocives contra els servidors.

1.3 Enfocament i mètode seguit

El concepte d'aplicació per comparar preus no es nou, tot i que normalment sol estar més enfocat que a serveis que no pas a productes. És per això que la única opció viable per elaborar l'aplicació ha sigut plantejar-la des del principi.

Cal tenir en compte que ha sigut vital per agilitzar el desenvolupament l'ús d'eines i llibreries de codi obert, sobre les quals se'n parlarà més endavant.

Per tal de donar agilitat al projecte i prioritzar les tasques necessàries per assolir les fites principals s'ha optat per desenvolupar fent ús de la metodologia RAD¹ lleugerament adaptada, ja que està pensada per treballar en grups de desenvolupament més grans.

Aquesta metodologia permet realitzar cicles de desenvolupament de curta durada (de 30 a 60 dies) i cadascuna d'aquestes iteracions està formada de següents fases:

- Planificació: És l'etapa inicial, consisteix en planejar quina part del projecte es dura terme, tenint en compte les prioritats i el temps que s'invertirà en finalitzar-la.

¹ Abreviatura de *Rapid Agile Development*

- Desenvolupament: En aquesta fase es dissenyen i implementen les tasques que componen el cicle.
- Test: S'avalua es compleixen els requisits especificats la planificació finalitzat sense fallades.
- Entrega: Es dona per finalitzada la iteració i es procedeix a iniciar un nou cicle.



Figura 1: Cicle desenvolupament àgil

1.4 Planificació del Treball

1.4.1 Recursos necessaris

Si assumim que una empresa hagués subcontractat aquest projecte tindria els següents costos:

Servei	Hores	Preu/h	Total
Gestió de Projecte	40	25 €	1000 €
Arquitectura	30	40 €	1200 €
Disseny	110	20 €	2200 €
Implementació	180	20 €	3600 €
QA	40	25 €	1000 €

Taula 1: Cost serveis

Recurs	Cost
Repositori Git (GitLab)	0 €
Allotjament de pàgines web (GitLab)	16 € / mes
Servidor Cloud (Amazon Web Services)	70 € / mes
Domini web	8 € /any
Certificat SSL	6 € /any

Taula 2: Cost recursos

En total el cost ascendiria a **9000 € de pressupost i 87,17 €/mes de manteniment.**

1.4.2 Planificació de tasques

Tasca	Inici	Fi	Durada
PAC 1: Pla de treball	20/02/2019	06/03/2019	15
Redacció del projecte	20/02/2019	26/02/2019	7
Elaboració del pla de treball	27/02/2019	06/03/2019	8
PAC 2: Disseny	07/03/2019	10/04/2019	35
Anàlisi d'eines necessàries	07/03/2019	10/03/2019	4
Anàlisi de desplegament	11/03/2019	12/03/2019	2
Instal·lació de l'entorn de treball	13/03/2019	13/03/2019	1
Disseny de l'Aplicació: Wireframes	14/03/2019	20/03/2019	7
Disseny de l'Aplicació: Maquetació	21/03/2019	03/04/2019	14
Modelació de la base de dades	04/04/2019	10/04/2019	7
PAC 3: Implementació	11/04/2019	29/05/2019	49
Client Web: Modul de components	11/04/2019	20/04/2019	10
Client Web: Modul de registre	21/04/2019	24/04/2019	4
Client Web: Modul de perfil	25/04/2019	29/04/2019	5
Servidor Web: rutes de la API	30/04/2019	01/05/2019	2
<i>Servidor Web: Models</i>	02/05/2019	04/05/2019	3
<i>Servidor Web: Controladors</i>	05/05/2019	10/05/2019	6
<i>Servidor Web: Scraping</i>	11/05/2019	22/05/2019	12
Desplegament: CI / CD	23/05/2019	26/05/2019	4
Desplegament: Certificat SSL	27/05/2019	29/05/2019	3
PAC 4: Memòria	30/05/2019	12/06/2019	14
Correccions	30/05/2019	03/06/2019	5
Finalitzar memòria del projecte	04/06/2019	10/06/2019	7
Elaboració de la presentació	11/06/2019	11/06/2019	1
Gravar vídeo defensa del treball	12/06/2019	12/06/2019	1

Taula 3: Planificació

1.4.3 Diagrama de Gantt

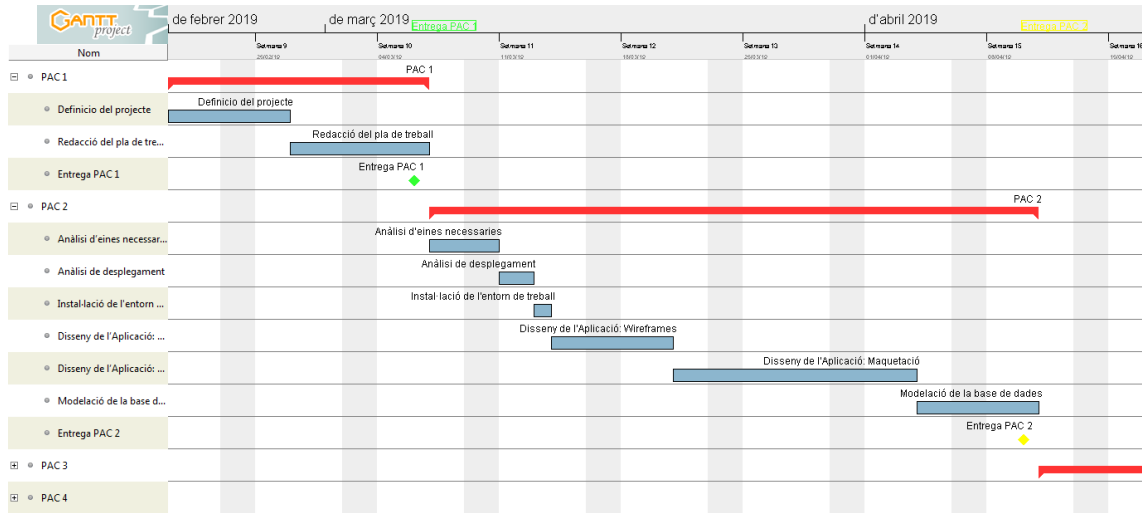


Figura 2: Diagrama de Gantt 1

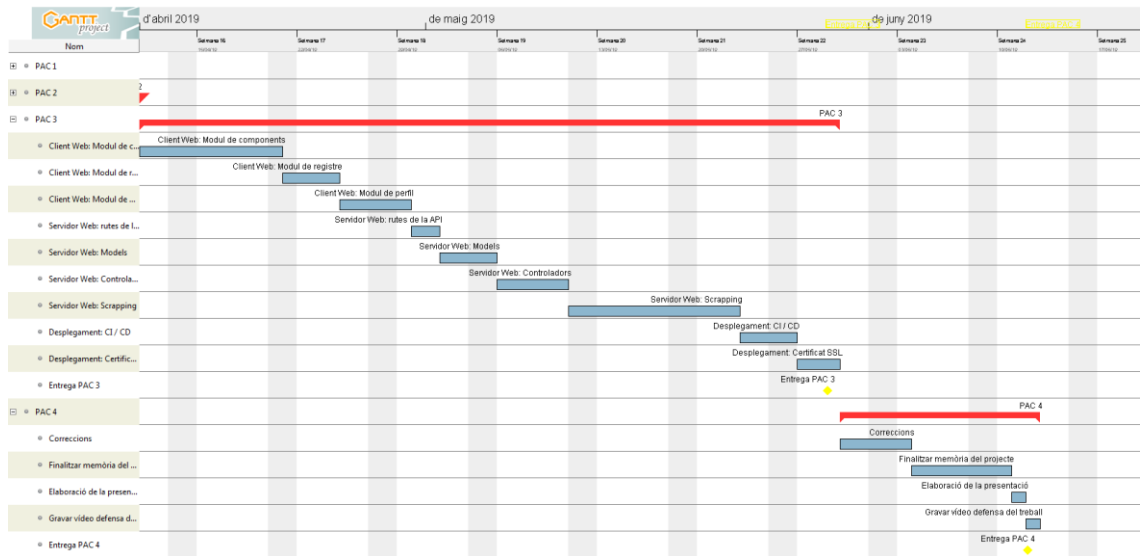


Figura 3: Diagrama de Gantt 2

1.5 Breu sumari de productes obtinguts

Com a resultat del treball, s'ha desenvolupat una aplicació web allotjada al núvol que compleix amb el producte mínim viable que estava estipulat en el pla de treball.

El portal disposa principalment de dos serveis per als usuaris:

- Per una banda, els usuaris poden accedir al llistat de productes que està present a la base de dades separat en diverses categories. Des de aquest llistat poden cercar el que producte que desitgin, o be, filtrar per les característiques per obtenir resultats mes específics. Des de el producte desitjat es pot navegar vista de detall, on es mostra tota la informació més detallada amb imatges, enllaços als diferents venedors, un gràfic amb l'evolució del preu en cadascun dels venedors i, per últim, una secció per a llegir comentaris dels altres usuaris o afegir-ne de nous.
- En segon lloc, els usuaris que hagin accedit amb el seu compte de la pagina, podran accedir des de el menú a la secció que permet construir el seu ordinador per peces. Afegint peces a traves del la vista del llistat de components, es va actualitzant la taula amb el preu total acumulat. Una vegada s'han afegit totes les peces, es dona la opció al usuari de finalitzar-la, que consisteix en bloquejar els elements seleccionats i publicar-ho a la vista de *builds*, on estaran totes les configuracions que els usuaris hagin publicat.

Per últim, els usuaris de tipus administrador podran afegir o modificar els productes a traves de la secció d'opcions de la vista de productes.

1.6 Breu descripció dels altres capítols de la memòria

En el segon bloc s'ofereix una explicació de l'arquitectura de l'aplicació, donant una visió global per al conjunt dels elements que la formen i també una visió específica per ajudar a entendre en profunditat cadascun d'aquests. També es justificaran les decisions preses en favor d'aquesta arquitectura.

El tercer bloc defineix el disseny del treball, on es detallen quines son les guies d'estil que s'han proposat, quins son els dissenys emprats en l'aplicació i, finalment, un esquema amb la navegació de la pagina web.

El quart bloc presenta les eines que s'han utilitzat al llarg del desenvolupament, així com també els llenguatges i llibreries sobre la que es recolza la implementació acompanyat d'una breu explicació de cadascuna..

El cinquè capítol conte un resum dels resultats obtinguts en forma de captures de pantalla i un estudi del rendiment de l'aplicació.

Al bloc numero sis, s'exposen les conclusions s'han obtingut amb el desenvolupament del projecte, la planificació i les metodologies

Els últims blocs contenen bibliografia amb les fonts citades o consultades per l'elaboració del treball, un glossari amb els termes tècnics emprats durant el document i annexos per complementar-lo.

2. Arquitectura del sistema

2.1 Arquitectura general

Buildy es una aplicació web moderna que consta dels tres components elementals en qualsevol aplicació web: El client web, el servidor web i la base de dades.

Per posar en sintonia aquests components existeixen diferents patrons d'arquitectura (Model-Vista-Controlador, Client-Servidor, *Serverless*...) i la que s'ha considerat més adient per el projecte és **l'arquitectura de 3 capes**.

Aquesta coneguda arquitectura consisteix en separar el conjunt del sistema en 3 capes, de manera que cada capa només pugui comunicar-se amb la capa adjacent. Les capes que la formen son:

- **Capa de presentació:** És la capa que es mostra el navegador, conte el conjunt de les interfícies gràfiques i és l'única en amb la que l'usuari pot interactuar directament.
- **Capa de negoci:** És la capa que conte la lògica de negoci, es a dir les funcionalitats internes de l'aplicació i que es comunica amb la capa de presentació proporcionant-li la informació pertinent.
- **Capa de persistència:** És la capa que s'encarrega de emmagatzemar les dades, normalment en forma de base de dades, i les serveix a la capa de negoci.

Els motius per haver triat aquesta arquitectura son principalment que l'aïllament de les capes permeten mantenir cada component del sistema desacoblat i cohesionat. Aquest aïllament de components, també facilita la possibilitat de fer un canvi de tecnologia, ja que no genera cap dependència de fer-ne servir una en concret o be escalar l'aplicació horitzontalment.

La infraestructura resultant del projecte es pot veure a continuació:

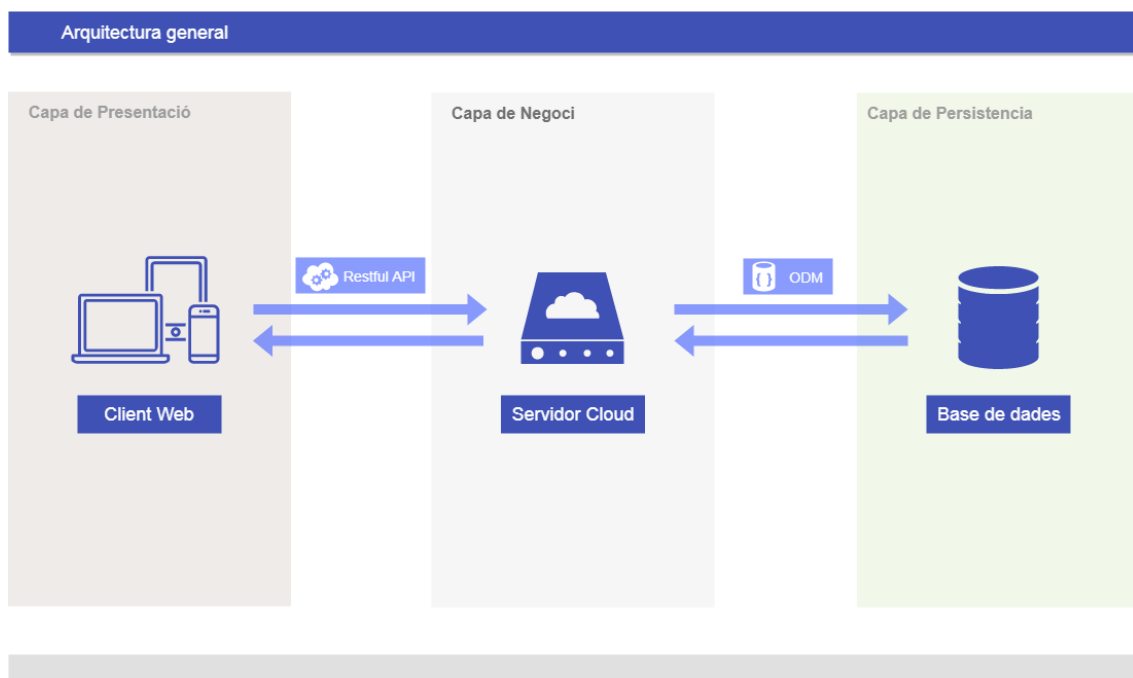


Figura 4: Arquitectura general

2.2 Arquitectura del client web

Un dels principals problemes del desenvolupament web tradicional es la redundància en el codi, ja que normalment es tendia a repetir codi en les diferents vistes (estructura, estils, funcionalitat...). Per tal maximitzar la reutilització de codi, facilitar manteniment i escalabilitat s'ha implementat una arquitectura que segueix el patró Model-Vista-VistaModel-Controlador (MVVM) i esta basada en components.

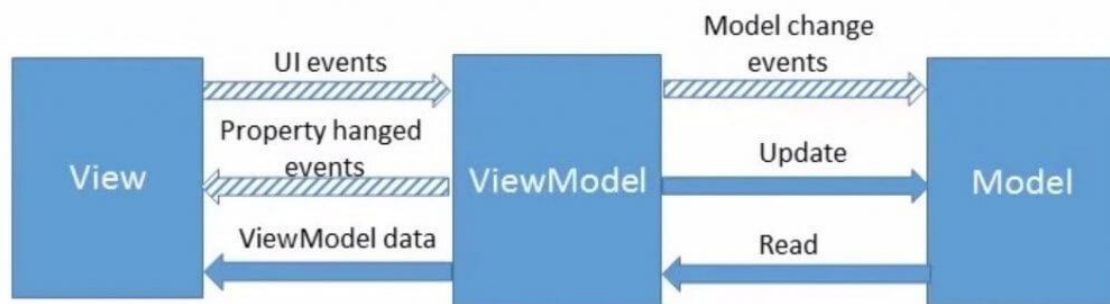


Figura 5: Patró MVVM

En aquesta arquitectura el codi esta estructurat en mòduls. Cada mòdul es independent i reutilitzable en altres mòduls mitjançant el patró d'injecció de dependències.

Un mòdul es un bloc de codi que conte un conjunt de components i serveis que es faran a l'aplicació sota demanda.

Els components son classes que agrupen la vista que es mostra en el navegador, els estils associats a aquesta vista i la lògica que se li aplicarà.

Els serveis son classes que contenen lògica o data que no esta directament relacionada amb un component en concret, per tant es pon injectar a qualsevol component.

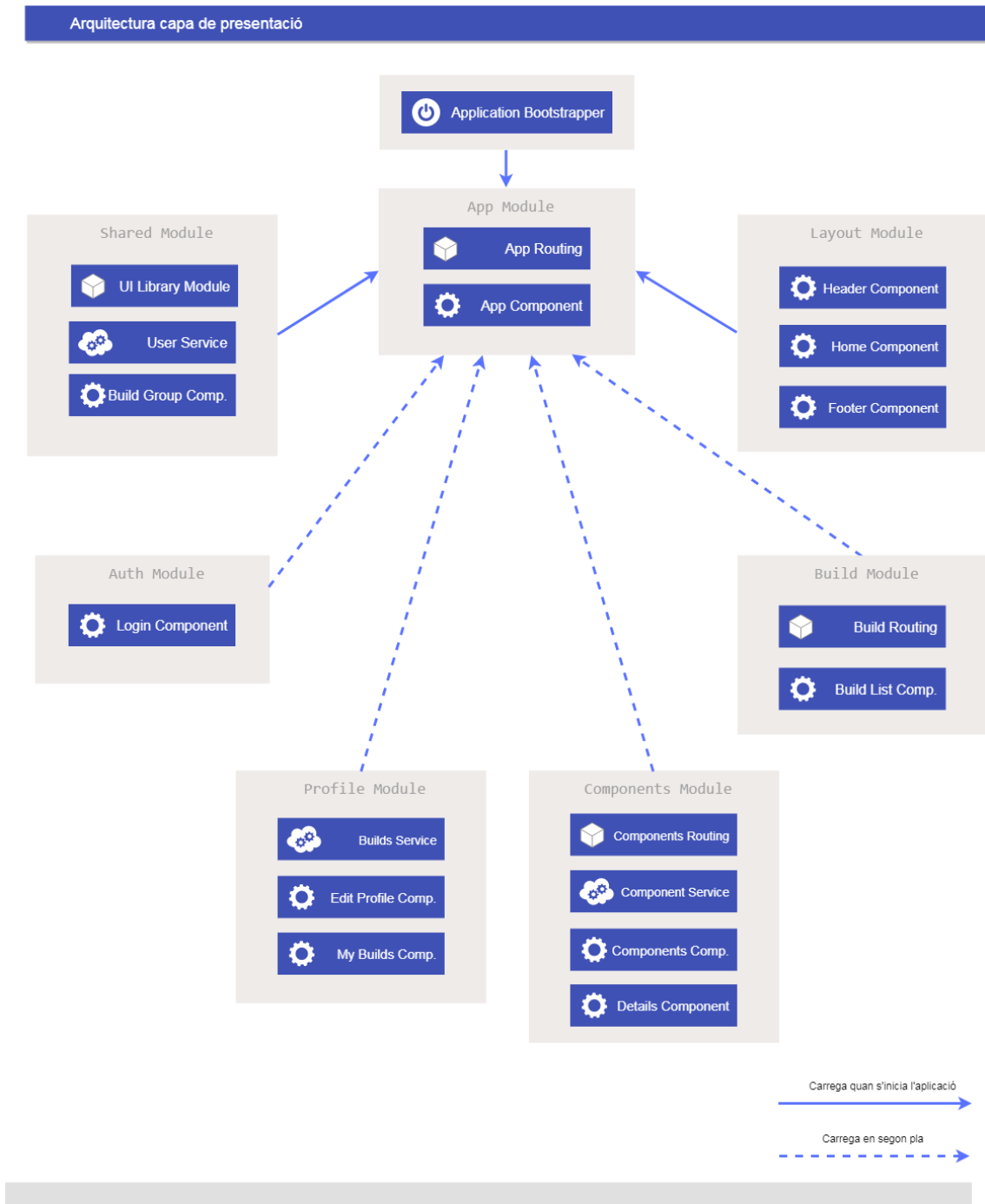


Figura 6: Arquitectura de la capa de presentació

2.2 Arquitectura del servidor

En capa de negoci no es segueix cap patró concret d'arquitectura, ja que els elements que la componen son una mica dispersos. Per això s'ha dissenyat una arquitectura semblant al que ens podríem trobar en un patró Model-Vista-Controlador o **MVC**, en aquest cas, però, no hi ha vista.

L'encaminador (**router**) actua com a patró de disseny de **façana**, es a dir, s'encarrega de assignar el controlador i la funció a executar d'aquest controlador segons el tipus de petició HTTP que rep el servidor.

El **middleware** s'encarrega de realitzar comprovacions abans de que s'executi la funció en el controlador, per exemple serveix per validar que l'usuari que ha fet la petició esta correctament autenticat, o be, per comprovar que l'objecte que s'ha enviat al cos de la petició compleix amb uns criteris de validació específics.

A més de la API REST, el servidor conte el mòdul de **scraping**, que roman separat de la resta de components però aprofita la connexió oberta amb la base de dades per fer les transaccions oportunes. Aquest mòdul es connecta a traves de un **servidor proxy** a les pagines de les quals en volem extreure la informació

Les transaccions en base de dades es fan a traves de un **ODM**² que assisteix a generar quèries contra la base de dades fent servir el paradigma de programació orientada a objectes.

² Sigles de *Object Data Modeling*

Arquitectura capa de negoci

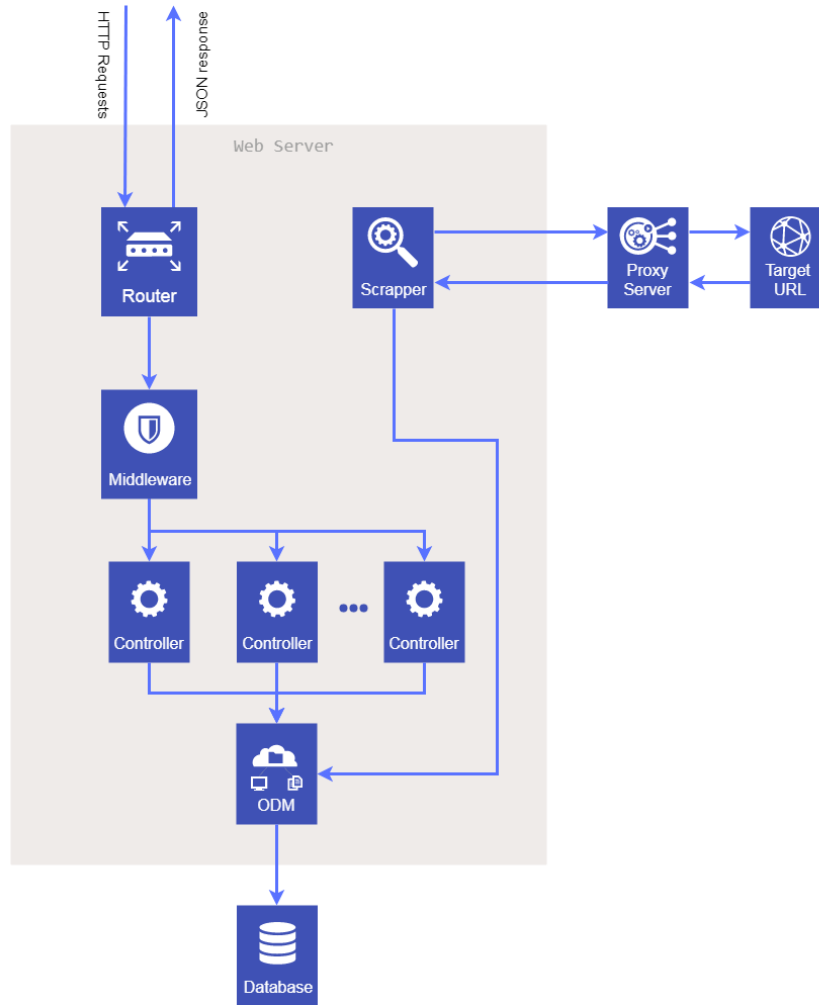


Figura 7: Arquitectura de la capa de negoci

2.3 Esquema de la base de dades

La capa de persistència fa ús d'una base de dades de tipus no relacional o **NoSQL**, això es degut a que el model de base de dades es bastant senzill i es més flexible, per tant permet anar evolucionant el model inicial a mida que es va desenvolupant sense tenir que realitzar tedioses migracions. A més ofereix una velocitat molt més elevada en *queries* senzilles i treballa nativament amb objectes **JSON**³.

A continuació es pot veure el esquema amb els objectes que s'ha modelat la base de dades:

³ Acrònim de *JavaScript Object Notation*

Diagrama de Classes

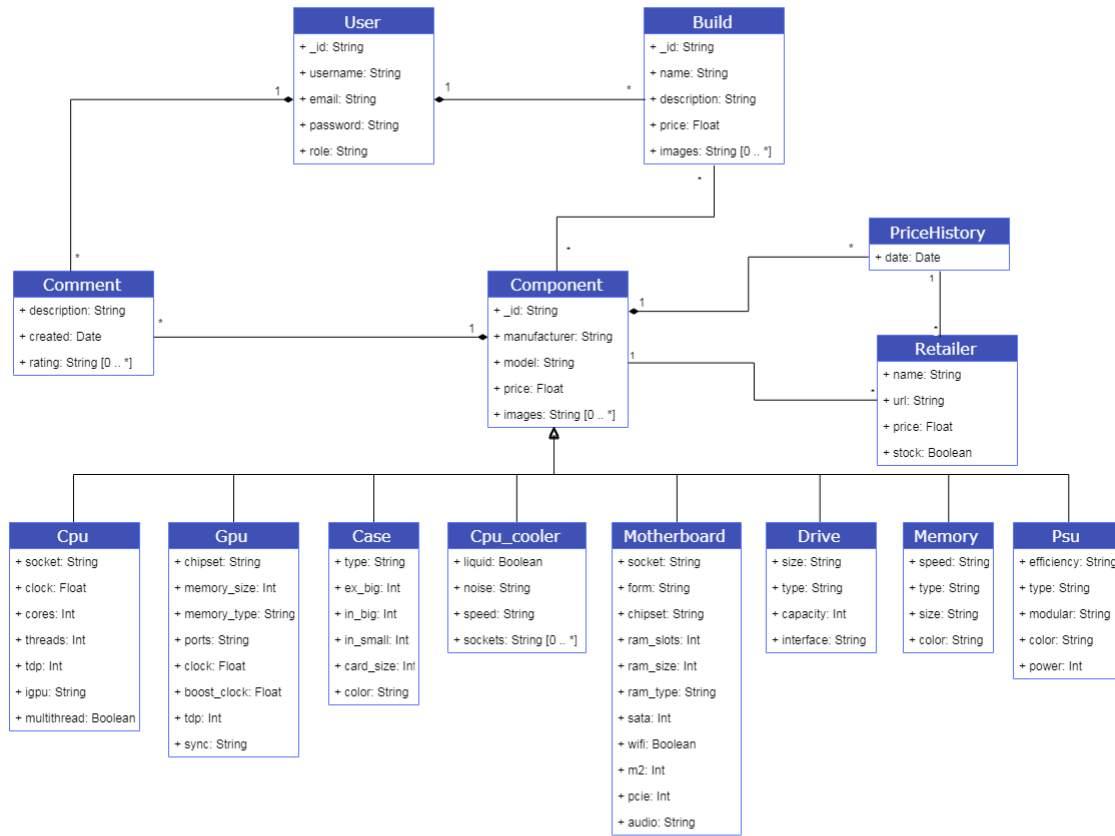


Figura 8: Diagrama de classes

2.4 Scraper

2.4.1 Que és el scrapping?

El *scrapping* consisteix en un procés automatitzat que extrau informació periòdicament d'unes pàgines webs determinades mitjançant peticions HTTP per obtenir, en aquest cas, els preus dels productes que posteriorment s'emmagatzemaran a la base de dades.



Figura 9: Scrapping

2.4.2 És legal?

Una de les primeres coses que et preguntes quan sens a parlar d'aquesta metodologia es la possibilitat d'estar incorrent en algun delictes.

En la majoria de països fora de la Unió Europea existeixen lleis que determinen que qualsevol informació pública en una pàgina web es subjecte de ser aprofitada per tercers. En el cas d'Espanya, com a membre de la UE, existeixen lleis com la recentment aprovada LOPD que regulen la protecció de dades i, juntament a la falta de una regulació per el scrapping de pàgines, fan que aquest cas sigui més ambigu.

Fent us de la jurisprudència, hi han agut en el últims anys algunes demandes per aquest mateix fet, com per exemple la querella que va posar l'empresa Ryanair a Atrápalo que el jutjat va desestimar⁴.

Respecte a l'aplicació, l'única informació que s'extreu son el preus i si hi han existències dels productes, cosa que esta lliure de cap protecció de dades.

2.4.3 Implementació

Hi ha diverses estratègies per implementar un *scraper*, per aquest projecte s'ha buscat la manera més eficient, en el sentit de reduir al màxim el nombre de peticions HTTP per tal de tenir el menor impacte possible amb el tràfic generat.

Per fer-ho enlloc de demanar la informació producte per producte, es fa una petició gran que retorna tot un llistat. Aquest llistat obtingut s'analitza i es compara producte per producte contrastant contra els productes de la base de dades.

Per fer la comparació de productes en primer lloc s'intenta fer un emparellament amb l'enllaç del producte, en aquest cas es sap segur que aquest es el producte en qüestió. En cas de que els productes de la base de dades no tinguin enllaços, s'intenta fent una comparació amb el nom i si es donen un conjunt de condicions en considera un emparellament vàlid i també es guarda el preu del producte.

En resum, es pot observar el flux de dades en el següent diagrama:

⁴ <https://supremo.vlex.es/vid/desleal-ryanair-atrapalo-as-411388894>

Diagrama de flux del scraper

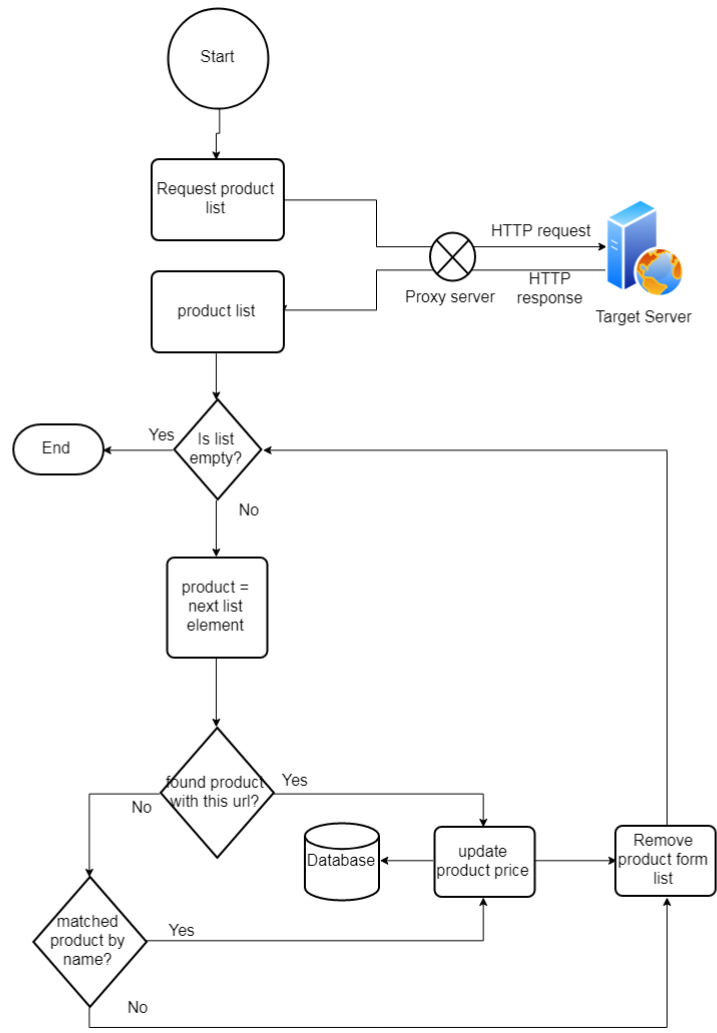


Figura 10: Diagrama de flux

3. Disseny

3.1 Guia d'estils

L'aplicació segueix una guia d'estils basada en el llenguatge de disseny establert per Google anomenat *Material Design*. Aquest sistema de disseny esta ideat per crear interfícies simples i minimalistes.

Per elaborar aquesta línia de disseny, els dissenyadors es van inspirar en les superfícies que trobem al mon real, intentant-ne reproduir al màxim la manera en que reflecteixen la llum i les ombres.

A més, en el últims anys s'ha expandit molt fins al punt de que pràcticament s'ha convertit en un estàndard d'aplicacions modernes, gracies també a la seva flexibilitat que permet implementar-se en qualsevol tipus d'aplicació.

3.1.1 Paleta de colors

Els colors oscil·len entre diverses tonalitats de blau, que donen una sensació agradable per llegir i queda bastant neutre.

Sobre el color principal es presenta el text de color blanc en cas de que la intensitat sigui mitja o elevada, o be, negre si es d'intensitat més suau.

La resta dels continguts estan sobre fons blanc i text negre.

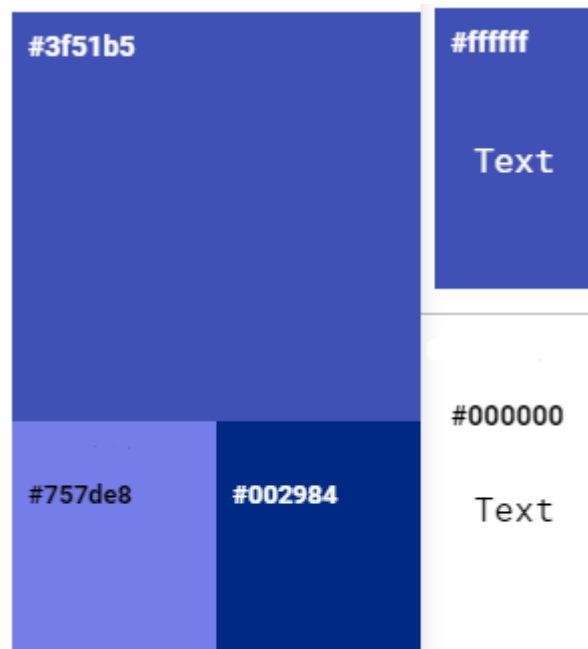


Figura 11: Paleta de colors

3.1.2 Logotip

El logotip es representa amb la mateixa paleta de colors que la pagina i presenta un disseny minimalista que combina be amb la resta del disseny.



Figura 12: Logotip

3.1.3 Tipografia

La font utilitzada en tota l'aplicació es Roboto, una font de gratuïta i de lliure us comercial basada en la tipus de lletra Sans-serif i creada el 2011 per Christian Robertson, pensada per ser moderna, però familiar i emocionant.

Roboto

Headline	Regular 24pt
Title	Medium 20pt
Subheader	Regular 16pt
Body 2 / Menu	Medium 14pt
Body 1	Regular 14pt
Caption	Regular 12pt
BUTTON	MEDIUM 14PT

Figura 13: Tipografia

3.1.4 Components

Els elements de la pagina, seguint la guia d'estil, presenten un aspecte senzill i minimalista, fent us de les ombres per donar la sensació d'elevació. Els components emprats son els següents:

Barra superior



Figura 14: Component barra superior

Pestanyes



Figura 15: Component pestanyes

Panell desplegable

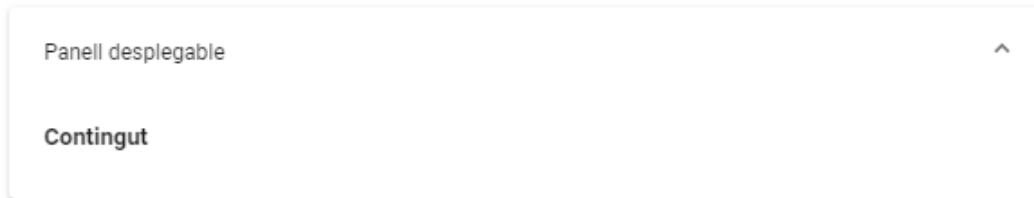


Figura 16: Component panell desplegable

Taula

Columna 1	Columna 2	Columna 3
Fila 1	Fila 1	Fila 1
Fila 2	Fila 2	Fila 2
Fila 3	Fila 3	Fila 3
100 total		

100 total << < 1 2 3 4 5 > >>

Figura 17: Component taula

Diàleg



Figura 18: Component diàleg

Notificació

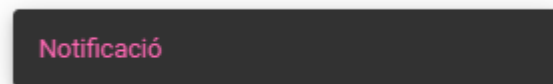


Figura 19: Component notificació

Formularis

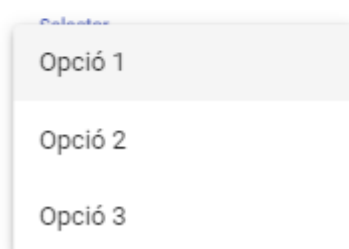
Input senzill



Àrea de text



Selector



Desmarcat Marcat

Opció 1 Opció 2

Botó normal

Botó Primari

Figura 20: Component formulari

3.2 Navegació

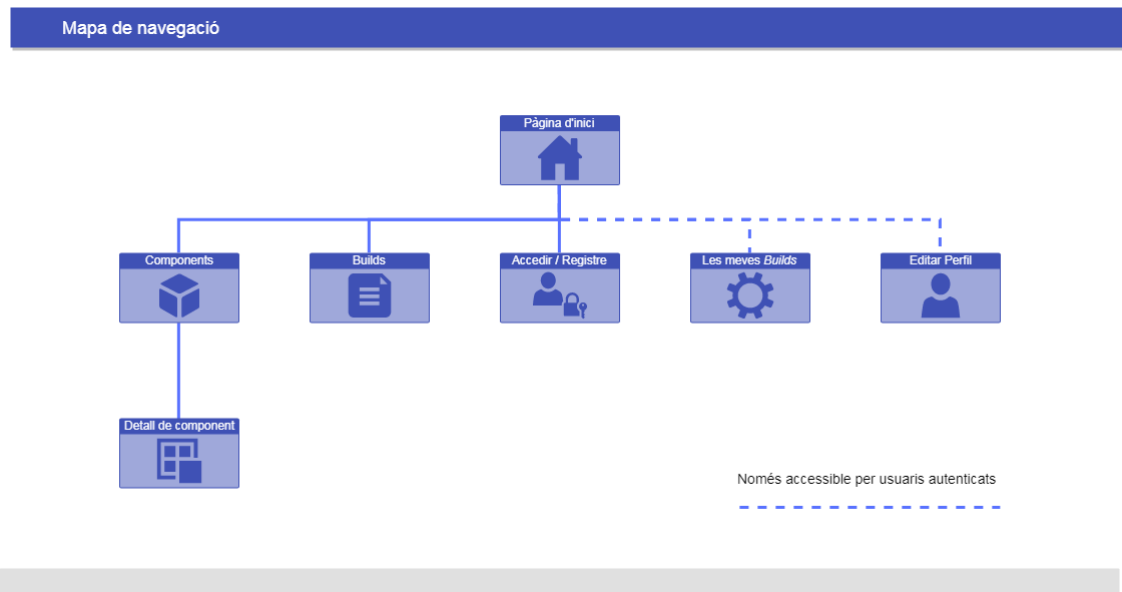


Figura 21: Navegació

4. Desenvolupament

Per tal de dur a terme els objectius plantejats, ha sigut imprescindible elegir les tecnologies adequades i també fer us d'eines que han ajudat a agilitzar el desenvolupament, així com proporcionar velocitat, fiabilitat i seguretat a l'aplicació.

4.1 Llenguatges de programació

Per a realitzar la maquetació web s'ha optat per fer servir HTML i SCS, a raó de que SCSS aporta un conjunt de avantatges respecte el CSS tradicional com son l'ús de variables, sintaxi amb regles niades, entra d'altres.

Per la lògica el llenguatge elegit ha sigut el Typescript. Aquest llenguatge es un superconjunt de JavaScript, es a dir que per fer-lo servir s'ha de compilar abans. Typescript presenta un gran nombre de beneficis, entra d'altres en destaquen:

- Al ser compilat, dona informació dels errors abans de la seva execució
- Millora el suport a la programació orientada a objectes aportant classes, interfícies, herències...
- Afegeix tipificació a les variables.

4.2 Eines de desenvolupament

El programari utilitzat en les diferents etapes de la implementació son les següents:

- **Visual Studio Code:** Editor de codi font multi plataforma desenvolupat per Microsoft. Està desenvolupat en codi obert, disposa d'eines per depurar el codi i un gran nombre d'extensions desenvolupades per la comunitat.
- **Git:** Programari de control de versions.
- **GitLab:** Plataforma web destinada a allotjar software que utilitza git. A més, permet desplegar pàgines web estàtiques de forma gratuïta i proporciona un gran nombre d'eines per al desenvolupament en grups.
- **Amazon Web Services:** Suite d'eines destinades al desenvolupament de tot tipus d'aplicacions. En concret s'ha fet us de l'eina Amazon Elastic Compute Cloud o **Amazon EC2**, que permet instanciar servidors dedicats en el núvol que romanen en funcionament les 24 hores del dia i accessibles via SSH. També el servei **Amazon Route 53** ha servit per gestionar el DNS del domini al servidor.
- **Robo 3T:** Interfície gràfica multi plataforma per visualitzar, consultar o modificar bases de dades basades en MongoDB
- **Gimp:** Editor d'imatges.
- **GanttProject:** Editor de diagrames Gantt de codi obert i multiplataforma.
- **Draw.io:** Editor de diagrames gratuït, forma part del conjunt d'eines de Google.
- **Microsoft Office:** Suite d'ofimàtica de Microsoft.

4.3 Llibreries i *Frameworks*

Les llibreries sobre les que es recolza el projecte son:

- **Angular:** *Framework* de codi obert desenvolupat en Typescript per Google destinat a construir aplicacions web orientades a components.
- **Node.js:** Entorn d'execució d'aplicacions en JavaScript amb arquitectura orientada a esdeveniments des del qual s'executa el servidor web i el *scapper*.
- **Express.js:** *Framework* escrit en JavaScript que s'executa sobre la plataforma de Node que serveix per construir APIs robustes i segures.
- **MongoDB:** Sistema gestor de bases de dades no relacionals.
- **Mongoose:** Llibreria ODM⁵ que proporciona un entorn per modelar els documents per a bases de dades MongoDB.

⁵ Acrònim d' *Object Data Modeling*

5. Resultats Obtinguts

Un cop finalitzat el desenvolupament s'han obtingut els resultats següents:

5.1 Pantalles i funcionalitats

En obrir l'aplicació ens mostra la *landing* principal, que conte una breu descripció dels serveix que ofereix al visitant. També es mostra, com a totes les pantalles, un menú superior per accedir a les diferents vistes.

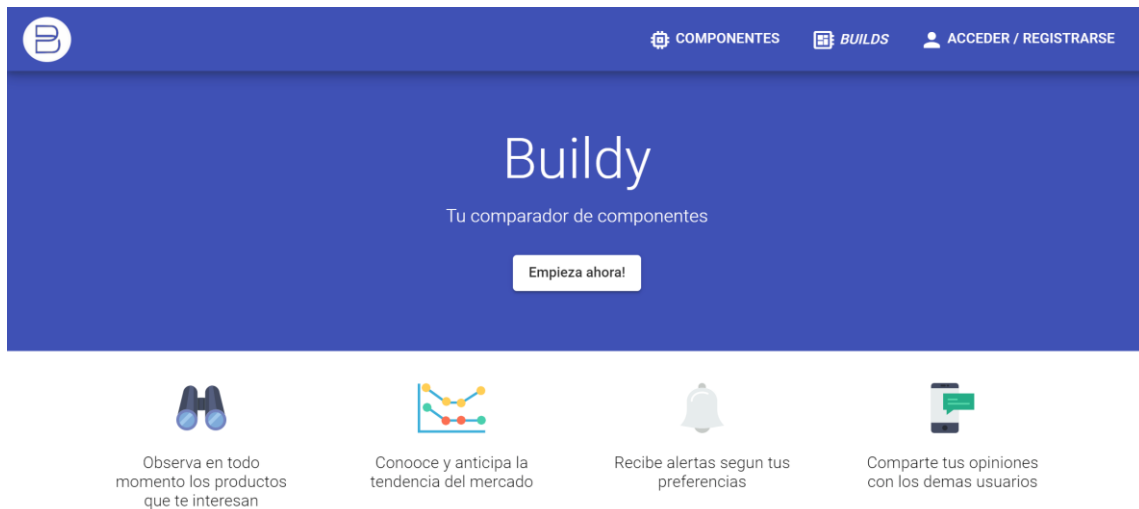


Figura 22: Landing principal

La primera acció que podem realitzar es accedir o registrar-nos al portal. Es pot fer des de qualsevol vista clicant al boto “accedir / registrarse” del menú que desplegarà un diàleg amb les 2 accions.

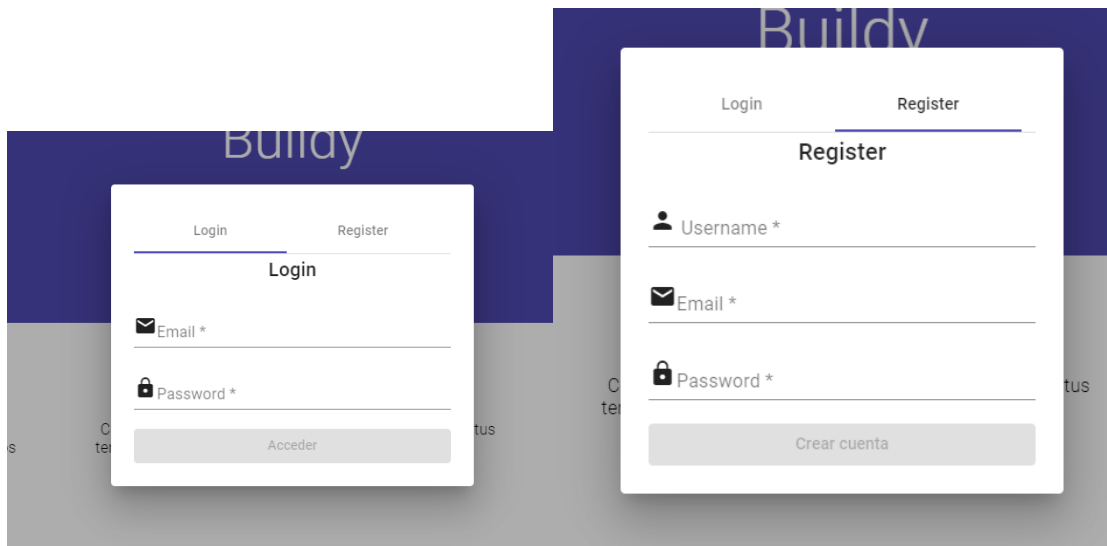


Figura 23: Diàleg registre

Si anem al menú “componentes” es desplegarà un llistat amb les diferents categories de components.

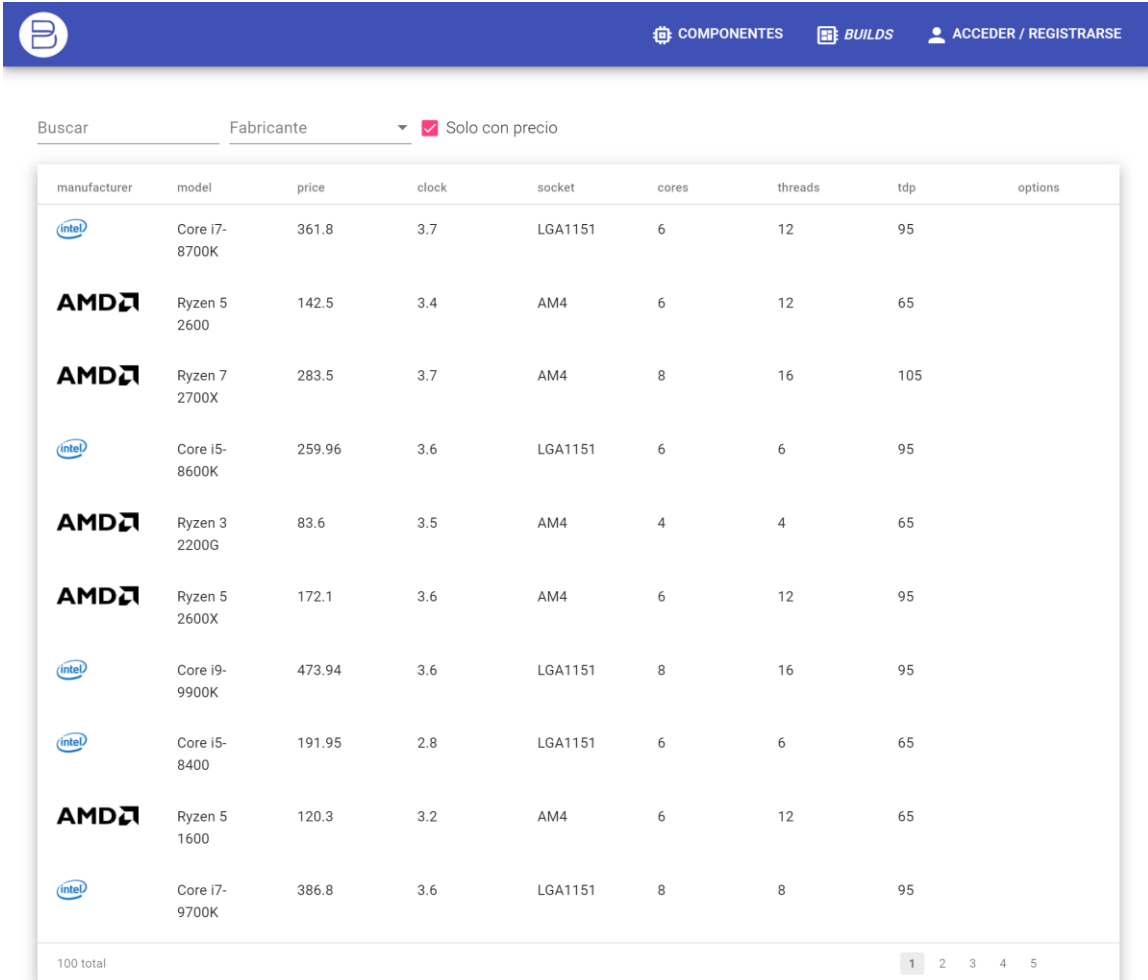


Figura 24: Menú componentes

I quan pressionem un d'aquests en portarà a la vista de llistat de components. Aquesta vista mostra una taula amb un resum dels productes d'aquella categoria.

Com hi han bastants productes s'ha pres la decisió d'implementar paginació per servidor, per tal de no sobrecarregar el client.

A més, la taula disposa de filtres per nom del model, el fabricant i oculta productes sense preu.



The screenshot shows a web interface with a blue header bar containing a search icon, navigation links for 'COMPONENTES', 'BUILDS', and 'ACCEDER / REGISTRARSE'. Below the header, there are search filters: 'Buscar', 'Fabricante', and a checked checkbox for 'Solo con precio'. The main content is a table with the following columns: manufacturer, model, price, clock, socket, cores, threads, tdp, and options. The table lists 10 products from Intel and AMD. At the bottom left, it says '100 total' and at the bottom right, there is a pagination control showing '1' as the active page and '2', '3', '4', '5' as other pages.














manufacturer	model	price	clock	socket	cores	threads	tdp	options
	Core i7-8700K	361.8	3.7	LGA1151	6	12	95	
	Ryzen 5 2600	142.5	3.4	AM4	6	12	65	
	Ryzen 7 2700X	283.5	3.7	AM4	8	16	105	
	Core i5-8600K	259.96	3.6	LGA1151	6	6	95	
	Ryzen 3 2200G	83.6	3.5	AM4	4	4	65	
	Ryzen 5 2600X	172.1	3.6	AM4	6	12	95	
	Core i9-9900K	473.94	3.6	LGA1151	8	16	95	
	Core i5-8400	191.95	2.8	LGA1151	6	6	65	
	Ryzen 5 1600	120.3	3.2	AM4	6	12	65	
	Core i7-9700K	386.8	3.6	LGA1151	8	8	95	

Figura 25: Taula de productes


Quan pressionem sobre alguna fila de la taula, navegarem fins a la vista de detall de component.

Aquesta vista mostra tota la informació detallada del producte, una galeria d'imatges, un apartat de comentaris, un llistat dels venedors que disposen d'aquest producte amb el seu preu i disponibilitat i per últim, l'historial de preu d'aquell producte per cadascun dels venedors.













COMPONENTES  BUILDS  ACCEDER / REGISTRARSE

Intel Core i7-8700K

Galeria

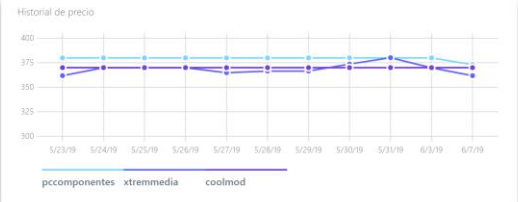


Información

-  manufacturer
Intel
-  model
Core i7-8700K
-  price
361.8
-  socket
LGA1151
-  clock
3.7
-  turbo
4.7
-  cores
6
-  threads
12
-  tdp
95
-  igpu
Intel UHD Graphics 630
-  multithreading
true

Fabricante	Precio	Stock
pccomponentes	372.99 €	true
xtremmedia	361.8 €	true
coolmod	369.95 €	true
3 total		

Historial de precio



Comentarios

Nuevo comentario




Figura 26: Detall de producte

Si retornem a la pagina de llistat de productes i accedim a la pagina com un usuari amb permisos d'administrador, veurem que ens permet afegir productes nous i modificar els que ja son al llistat.

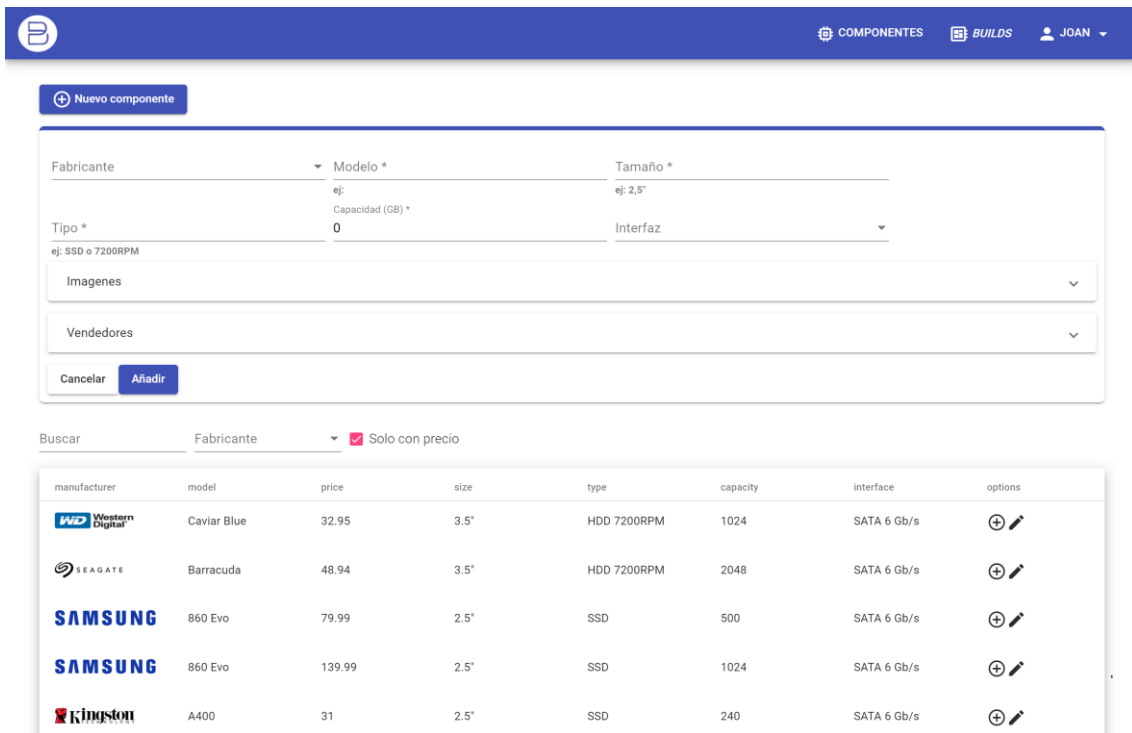


Figura 27: Nou component

Des de el menú d'usuari a la cantonada dreta, podem accedir a la pagina "mis builds". Aquí l'usuari pot fer la seva simulació de compra de ordinador per peces i calcular-ne el preu més econòmic basant-se els preus de l'aplicació.

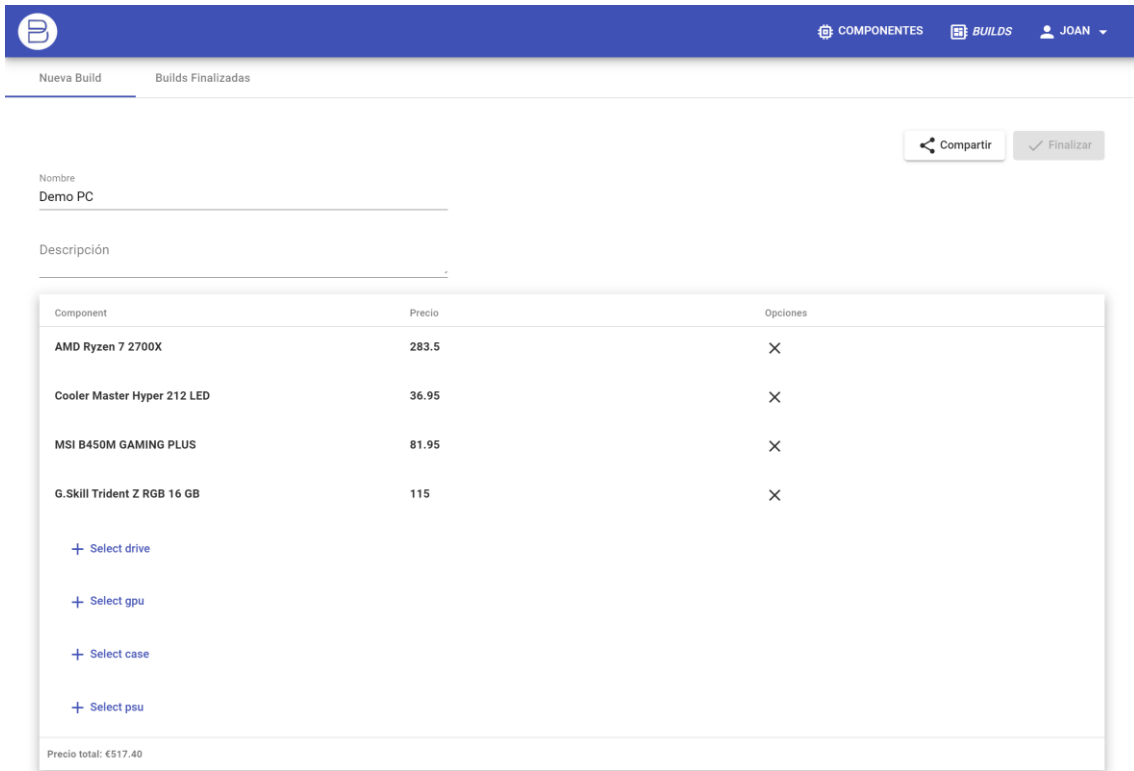


Figura 28: Zona usuari

Una vegada seleccionats tots els components l'usuari podrà clicar a finalitzar, llavors quedarà enregistrar l'ordinador a la base de dades. L'usuari podrà veure tots els seus ordinadors des de la pestanya "builds finalizadas".

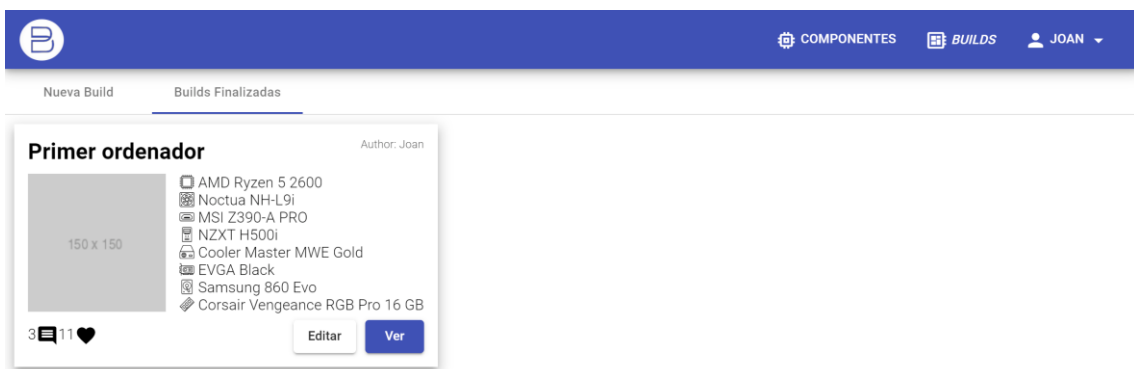


Figura 29: Builds finalizadas

Per últim, a través de la opció "builds" del la barra de menú superior podem veure les construccions que han fet tots els usuaris.

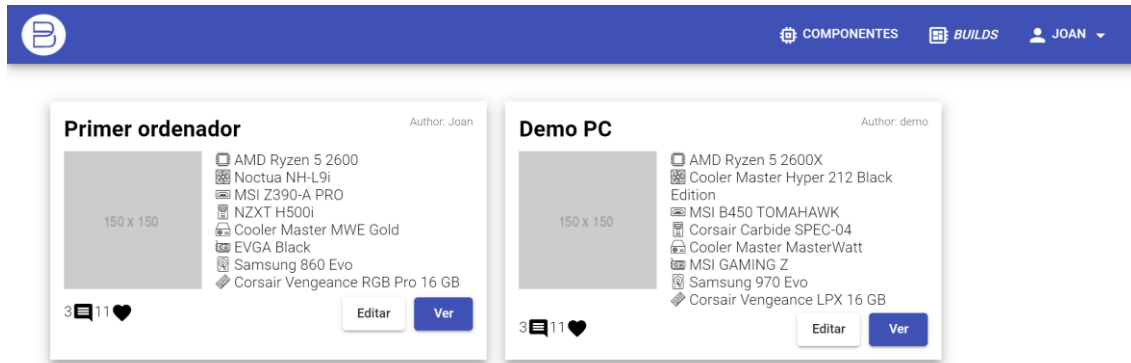


Figura 30: Vista builds

5.2 Rendiment

S'ha avaluat la pagina en diferents proves de rendiment (o *benchmarks*) i els resultats obtinguts son notòriament positius, tant per els temps de carrega com per l'ús de bones practiques. El principal punt negatiu es la falta de servir els recursos en cache, però això no es possible allotjant la pagina a GitLab.

Lighthouse

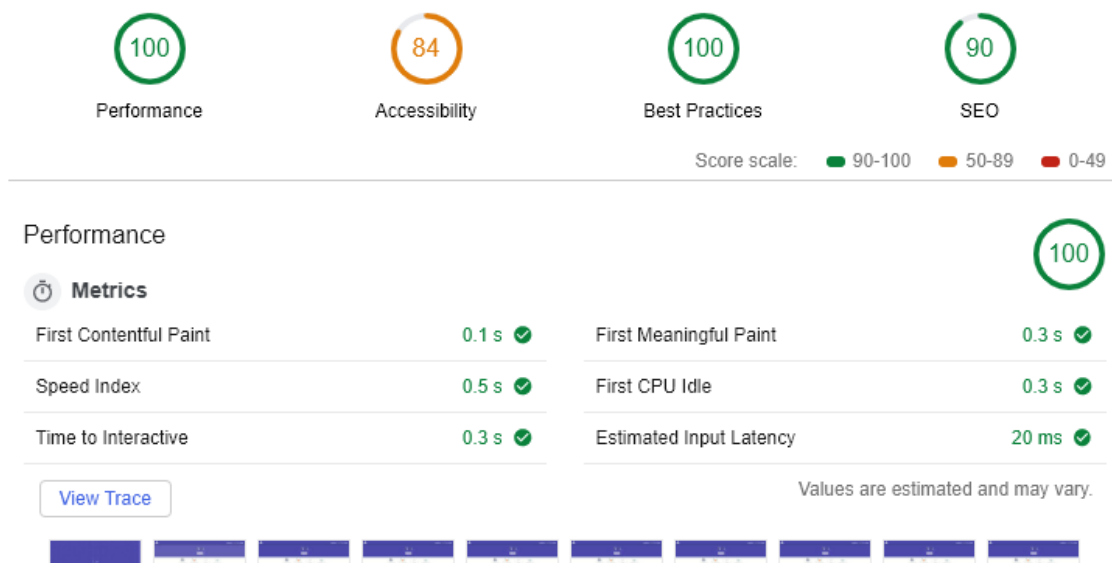


Figura 31: Prova rendiment Lighthouse

Webpagetest

Web Page Performance Test for

<https://pheder.gitlab.io/buildy>

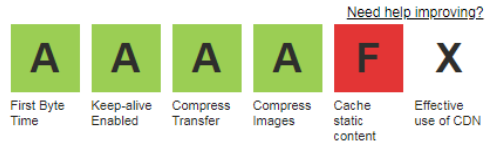
From: Dulles, VA - Chrome - Cable
12/6/2019 20:38:51

Tester: VM04-01-172.16.20.198

First View only

Test runs: 3

Re-run the test



Performance Results (Median Run)

	Load Time	First Byte	Start Render	User Time	Speed Index	Last Painted Hero	First Interactive (beta)	Document Complete			Fully Loaded			
								Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 2)	1.852s	0.245s	0.700s	0.737s	1.056s	2.000s	> 0.639s	1.852s	19	521 KB	1.954s	20	536 KB	\$8---

Figura 32: Prova rendiment Webpagetest

GTmetrix



Latest Performance Report for:

<https://pheder.gitlab.io/buildy>

Report generated: Wed, Jun 12, 2019 11:43 AM -0700

Test Server Region: Vancouver, Canada

Using: Chrome (Desktop) 62.0.3202.94, PageSpeed 1.15-gt1.2, YSlow 3.1.8

Performance Scores

PageSpeed Score A (93%) ^	YSlow Score C (79%) ^
-------------------------------------	---------------------------------

Page Details

Fully Loaded Time 2.0s ^	Total Page Size 537KB ^	Requests 20 ^
------------------------------------	-----------------------------------	-------------------------

Figura 33: Prova rendiment GTmetrix

6. Conclusions

El treball final de grau es la culminació d'un intens camí de formació, no es estrany que els alumnes es posin un gran nivell d'exigència per poder demostrar que han pogut fer seus tots aquests coneixements. En el meu cas particular ho he volgut transmetre a partir d'un projecte que combines la capacitat d'autoaprenentatge per elaborar alguna cosa diferent al que he fet durant aquests anys i també que fos un tema d'interès personal perquè esdevingués satisfactori a nivell personal.

Estic convençut de que ha sigut una experiència totalment productiva sigui quin sigui el resultat final, ja que penso que he consolidat la capacitat d'autogestionar-me a l'hora de portar un projecte en tots els seus estadis, des de el plantejament fins l'entrega.

Un altre motiu de satisfacció és que he tingut l'oportunitat de aprendre en matèries que no havia tingut durant el grau, com es el cas de les operacions de desenvolupament o DevOps, que m'han permès formar-me en el desplegament d'aplicacions en el núvol, adquisició i instal·lació de certificats SSL, desplegament continu, gestionar servidors proxy.... També he pogut indagar en el disseny d'interfícies i l'experiència d'usuari en pàgines web.

Respecte el objectius, durant la primera fase de plantejament de treball em vaig plantejar molts objectius, de fet, masses si volia aconseguir entregar-la a la data estipulada, es per això que vaig establir un llistat de objectius prioritaris que eren el que tenien que sortir per a completar un producte mínim viable per el treball i un llistat d'objectius no tant important però que estaria molt be que estiguessin per donar més sensació de producte complert.

Puc afirmar que els objectius principals els he pogut completar, però alguns dels secundaris s'han quedat a mig fer o estan poc provats i no s'han pogut afegir al projecte final. Això es degut principalment a dos factors, la inevitable

limitació del temps, que, encara que estiguin estimades les hores que li podria dedicar al projecte, no sempre ha pogut ser així. També el fet d'aprendre tecnologies noves m'ha resultat difícil d'estimar, perquè es complicat predir la corba d'aprenentatge d'una tecnologia que no has tocat mai.

En quant al calendari establert, crec que ha sigut bastant encertat, ja que al final del projecte s'han entregat totes les tasques i no s'ha tingut que fer cap canvi d'abast. És cert que he tingut que acabar algunes tasques fora de les entregues establertes, però al final penso que també es important tenir capacitat per detectar quan un projecte esta sortint-se de la planificació establerta i saber com redreçar-la amb la major brevetat possible.

Les línies de futur del projecte son ben amples, hi han moltes ampliacions que m'agradaria fer per donar més possibilitats als usuaris.

Per una banda m'agradaria oferir als membres la possibilitat de compartir a les xarxes socials els ordinadors que s'hagin construït a la pagina i també donar l'oportunitat de que afegixin fotografies en cas de que comprin els productes.

A més, seria interessant tenir una secció per estadístiques, que per una banda mostres la tendència dels preus per cada categoria de productes i per un altre en quins productes s'interessen més els usuaris, ja esta molt de moda el tema del *Big Data*.

En definitiva, el mon de les aplicacions web i els comerços electrònics esta evolucionant molt darrerament, així que de ben segur que apareixeran noves tecnologies i arquitectures que permetran perfeccionar-la.

7. Glossari

User Experience (UX): Fa referència a les emocions i actituds del usuari davant d'un producte o servei.

Framework: És un programari compost de diverses llibreries i classes que té com a objectiu servir de base per desenvolupar una aplicació.

RAD: Acrònim de "Rapid-application development". És una metodologia de desenvolupament àgil basada en cicles.

DevOps: Conjunt de practiques en l'enginyeria de software destinades a automatitzar o monitoritzar els processos de construcció, integració o proves d'aplicacions.

Servidor proxy: És tracta d'un servidor que actua com a intermediari en una connexió.

Big Data: Son aplicacions que concentren i tracten grans volums de dades, normalment enfocades en l'àmbit científic o estadístic.

Benchmark: En l'àmbit de la informàtica, son proves per determinar mètriques, com ara el rendiment o les bones practiques.

8. Bibliografia

What is Agile Software Development? [en línea]. GoodFirms [data de consulta: 02 de Juny de 2019].

Disponible a: www.goodfirms.co/glossary/agile-software-development/

Web Scraping: ¿legal o ilegal? [en línea]. Ecija [data de consulta: Març de 2019]. Disponible a: <https://ecija.com/web-scraping-legal-ilegal/>

Multitier architecture [en línea]. Wikipedia [data de consulta: Març de 2019].

Disponible a: https://en.wikipedia.org/wiki/Multitier_architecture

9. Annexos

Annex 1: Portal i instal·lació local

En el moment de presentar aquest treball, la pàgina web es troba allotjada en l'enllaç <https://pheder.gitlab.io/buildy/>.

Des d'aquest portal es poden provar les funcionalitats implementades que s'han comentat en el document. Si pel contrari es vol realitzar un desplegament en la maquina local, cal seguir aquest procediment.

Requeriments previs

En primer lloc cal instal·lar el programari necessari per dur terme la instal·lació i cal tenir en compte que la versió ha de ser igual o superior a la esmentada en cada cas.

Començar descarregant i executant [Node.js](#) en la seva versió 10 o superior, a més de l'entorn d'execució inclou un gestor de paquets necessari per descarregar les dependències.

A continuació cal instal·lar la versió 4 o superior del sistema gestor de bases de dades [MongoDB](#) seguint les instruccions que ens proporciona la pagina oficial.

Per últim cal executar la següent comanda des de el terminal “npm install -g @angular/cli” per descarregar la interfície de comandes per Angular.

Opcional: Es recomanable per obtenir la ultima versió del codi descarregar el programari de control de versions [Git](#) en la seva versió més recent.

Descàrrega del codi font

El codi font de l'aplicació es pot obtenir per dues vies:

- Annexionat a la memòria del projecte podem trobar el fitxer comprimit amb tot el codi necessari per construir el projecte.
- Descarregant els fitxers directament del repositori públic mitjançant Git. Aquest repositoris es troben en [aquest enllaç](#).

Per més informació del contingut del codi consultar l'annex "Estructura del projecte".

Instal·lació de les dependències

Abans d'executar, però, s'han d'instal·lar les dependències necessàries. Per fer-ho, cal obrir el terminal de comandes des de l'arrel de cadascuna de les carpetes del client i el servidor i escriure la comanda "npm install".

Execució i construcció del projecte

Per executar el servidor web, cal navegar amb el terminal fins la carpeta del servidor "buildy-server" i executar la comanda "npm run pre".

Pel client hi han dues alternatives:

- Si es vol executar el projecte sense generar fitxers cal introduir la comanda "ng serve - o". Un cop es completi s'obrirà una finestra al navegador predeterminat amb l'aplicació.
- Alternativament, si l'objectiu és generar els fitxer per exemple per fer un desplegament cal construir el projecte amb la comanda "ng build". Després d'executar-la, és pot observar que s'ha creat una carpeta anomenada "build" que contindrà tots els fitxer de l'aplicació ja compilats.

Annex 2: Estructura del projecte

Dins del fitxer comprimit del projecte trobarem les següents carpetes:

- **wireframes and captures:** Conte documentació gràfica en forma de captures de pantalla i maquetes del disseny.
- **buildy-client:** Inclou tots els fitxers del client web del projecte. Cal destacar-ne la carpeta **src**, que conte el codi font, el fitxer **angular.json**, que es l'arxiu de configuració principal del *framework* Angular i el fitxer **.gitlab-ci.yml** que es l'script de desplegament continu en l'entorn de GitLab.
- **buildy-server:** Inclou els fitxers del servidor i el *scraper*. A l'hora de instal·lar l'aplicació localment s'ha de generar un fitxer de configuració anomenat ".env" que no esta inclòs per defecte, però si un d'anomenat **".env.default"** que conte la configuració per defecte, tot i que es recomana canviar-la. També disposa de la carpeta **seeds**, que conte fitxers JSON amb informació per poblar una base de dades nova pel projecte (no es requerit però serveix per tenir una base de dades inicial).