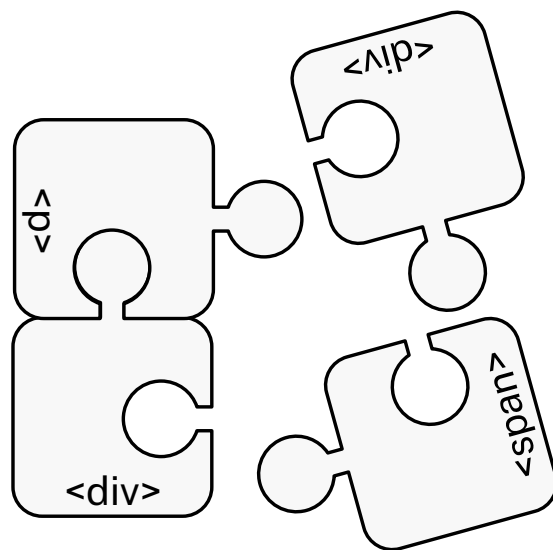


Grau en Multimèdia, UVic - UCC

WeStudioo

Treball de Fi de Grau



Autor/a: Maria Güell Molist

Tutor/a: Raymond Lagonigro Bertran

Vic, Juny de 2020

RESUM

- **Títol:** WeStudioo
- **Autora:** Maria Güell Molist
- **Tutor:** Raymond Lagonigro Bertran
- **Data:** Juny de 2020
- **Paraules clau:** HTML, aplicació web, Laravel

La idea d'aquest projecte és desenvolupar una eina per crear el codi font d'estructures HTML sense la necessitat de tenir nocions d'HTML. L'eina en qüestió està allotjada en un servidor web de manera que els usuaris hi podran accedir remotament a través d'internet.

L'objectiu final és obtenir una eina fàcil d'utilitzar que aconsegueixi resultats adequats. Concretament, les prestacions necessàries són: gestió d'usuaris, emmagatzematge al núvol, historial de revisions, comptabilitat amb llibreries CSS i generador de codi.

Cada usuari tindrà un espai privat on guardar els seus documents i un l'historial de revisions que recollirà tots els canvis que es realitzen als fitxers i l'opció de revertir-los.

Aquesta eina només permet el desenvolupament de codi referent a l'estructura dels elements d'un web (HTML), tanmateix, per tal de crear estructures complexes i visualment potents, cal vincular una llibreria d'estils. L'objectiu d'aquest projecte no és l'edició d'estils, però sí que contempla ser compatible amb diverses llibreries d'estils.

En aquesta primera fase del desenvolupament s'ha aconseguit un prototip funcional que permet crear estructures HTML de manera intuïtiva.

SUMMARY

- **Title:** WeStudioo
- **Author:** Maria Güell Molist
- **Supervisor:** Raymond Lagonigro Bertran
- **Date:** June of 2020
- **Keywords:** HTML, web application, Laravel

The idea of this project is to develop a tool to create the source code of HTML structures without the need to have notions of HTML. The tool in question is hosted on a web server so that users can access it remotely via the internet.

The ultimate goal is to get an easy-to-use tool that achieves the right results. Specifically, the necessary benefits are user management, cloud storage, revision history, accounting with CSS libraries, and code generator.

Each user will have a private space to save their documents and a review history that will collect all the changes made to the files and the option to revert them.

This tool only allows the development of code regarding the structure of the elements of a website (HTML), however, to create complex and visually powerful structures, it is necessary to link a library of styles. The aim of this project is not to edit styles, but it does intend to be compatible with various style libraries.

In this first phase of development, a functional prototype has been achieved that allows you to create HTML structures intuitively.

ÍNDEX

Introducció.....	6
Contextualització.....	6
Estudi de mercat (Benchmark).....	7
Objectius del projecte.....	9
Interfície gràfica per generar codi	10
Registre d'usuaris.....	10
Historial de versions.....	10
Múltiples llibreries CSS.....	10
Galeria multimèdia	11
Planificació	11
Etapas del projecte.....	11
Planificació temporal	11
Anàlisi inicial dels costos.....	12
Tecnologies utilitzades	13
Frameworks de desenvolupament	13
Software.....	13
Desenvolupament del projecte	14
Disseny	14
Aplicació web bàsica.....	17
CRUD de fitxers i revisions.....	17
Emmagatzematge de components	18
Estructura bàsica client de la pàgina Studio.....	18
Accessibilitat client de la pàgina Studio.....	18
Elements dins d'elements	19
Penjar imatges.....	19
Múltiples llibreries CSS.....	19
Opció de descàrrega.....	19
Historial de revisions.....	19
Arquitectura de l'aplicació.....	20
Servidor (Laravel).....	20
Client (Vue.js).....	21
Resultats.....	22

Portada.....	22
Gestió d'usuaris.....	22
Tauler usuari.....	24
Create	25
Studio	26
Media.....	27
Discussió i treball futur.....	27
Conclusions.....	28
References	29
Annexos.....	30
Benckmark: taula comparativa	30
Planificació temporal: Diagrama de Gantt.....	31

ÍNDIX DE FIGURES

Il·lustració 1 Esquema del funcionament d'una pàgina web.....	6
Il·lustració 2 Esquema dels fitxers d'una web	7
Il·lustració 3 Esquema de l'arquitectura client servidor.....	13
Il·lustració 4 Esquema de la distribució bàsica de l'aplicació.....	14
Il·lustració 5 Esquema de la distribució de la pàgina de portada	15
Il·lustració 6 Esquema de la distribució de la pàgina /studio	15
Il·lustració 7 Esquema de la distribució de la pàgina de taulell d'usuari.....	16
Il·lustració 8 Esquema de la distribució de la pàgina d'història.....	16
Il·lustració 9 Imatjotip del web	17
Il·lustració 10 Esquema dels components de vuetjs	21
Il·lustració 11 Portada del web (/).....	22
Il·lustració 12 Pàgina de registre d'usuaris (/register)	23
Il·lustració 13 Pàgina d'autenticació d'usuaris (/login)	23
Il·lustració 14 Portada després d'autenticar l'usuari (/)	24
Il·lustració 15 Tauler de l'usuari (/dashboard).....	24
Il·lustració 16 Història de revisions d'un fitxer (/history)	25
Il·lustració 17 Formulari per crear un fitxer (/create).....	26
Il·lustració 18 Pantalla studio (/studio).....	26
Il·lustració 19 Pantalla de la Galeria Multimèdia (/media).....	27

GLOSSARI

Client: és un dispositiu que consumeix el servei d'un servidor.

CRUD (Cread Read Update Delete): es refereix a les funcions *crea, llegeix, actualitza i suprimeix* que són les funcions bàsiques d'emmagatzematge persistent.

CSS (Cascading Style Sheets): llenguatge de fulls d'estils popularment utilitzat per definir l'aspecte i format d'un document HTML.

Drag and drop: tipus d'interacció que consisteix en l'acció d'arrossegar i deixar anar al damunt d'un objecte receptor, amb el ratolí.

JavaScript: llenguatge de programació web popularment utilitzat per costat de servidor.

HTML (Hypertext Markup Language): llenguatge de marcat popularment utilitzat per definir l'estructura dels continguts d'una pàgina web.

PHP (Personal Home Page): llenguatge de programació web popularment utilitzat per costat de servidor.

Servidor: dispositiu capaç de rebre peticions d'altres màquines i donar-hi les respostes adequades.

Web: conjunt de fitxers units per mitjançant enllaços. Normalment, allotjat en servidor remot.

WYSIWYG (What You See Is What You Get): interfície gràfica que permet que el que es veu durant l'edició d'un fitxer correspongui amb el resultat final.

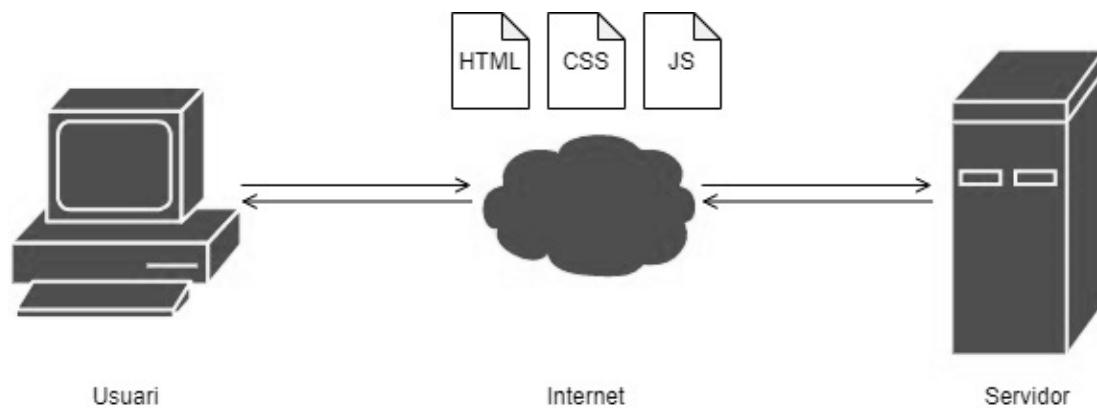
INTRODUCCIÓ

El projecte WeStudioo és un projecte web per generar estructures de codi HTML des d'una interfície gràfica per facilitar la feina als usuaris que necessitin crear continguts web però no tinguin coneixements de llenguatge web.

WeStudioo serà una web de manera que els usuaris hi podran accedir i utilitzar l'eina sense necessitat d'instal·lar cap software al seu ordinador.

CONTEXTUALITZACIÓ

Una web està formada per diversos fitxers allotjats a un ordinador (anomenat normalment servidor) que seran renderitzats pel navegador que l'usuari utilitzi. Està estandarditzat que els formats d'aquests fitxers sigui HTML per l'estructura del contingut, CSS pels estils i JavaScript per lògica.



IL·LUSTRACIÓ 1 ESQUEMA DEL FUNCIONAMENT D'UNA PÀGINA WEB

HTML (HyperText Markup Language) és el llenguatge per definir l'estructura del contingut del web. És important recalcar que només defineix l'estructura i no l'aparença d'aquest. Normalment una web sol estar formada per diferents pàgines i cada pàgina estarà definida per un fitxer HTML. Segurament, hi haurà un full d'estils que servirà per a tots els fitxers HTML. Per aquest motiu, en aquest projecte ens centrarem a generar HTMLs i no en els estils.

L'HTML és un llenguatge d'etiquetes de manera que cada element està delimitat per la seva etiqueta. L'estructura de cada element sol ser la següent:

```
<nom-de-etiqueta>Contingut de l'element</nom-de-etiqueta>
```

Cada element pot tenir certs atributs que s'especifiquen dins de l'etiqueta d'obertura:

```
<nom-de-etiqueta nom-atribut="valor atribut">Contingut de l'element</nom-de-etiqueta>
```

Dins de cada element hi pot haver altres elements:

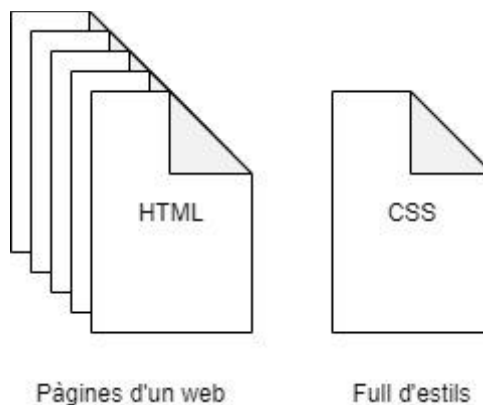
```
<article>
  <h1>Títol</h1>
```

```
<p>Cos</p>
</article>
```

Les pàgines HTML tenen una etiqueta anomenada head que conté totes les propietats de la pàgina i una etiqueta body amb tot el contingut de la pàgina.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Títol de la pàgina</title>
</head>
<body>
  <article>
    <h1>Títol</h1>
    <p>Cos</p>
  </article>
</body>
</html>
```

De manera que el full d'estils és quelcom que es defineix quan es desenvolupa el web i no requereix manteniment si no es desitja redissenyar l'estètica d'aquest. Tanmateix, les pàgines de contingut se solen modificar freqüentment.



IL·LUSTRACIÓ 2 ESQUEMA DELS FITXERS D'UNA WEB

Molts gestors de contingut web com *Wordpress* o *Drupal* són eines molt centrades a crear i mantenir un web per usuaris sense expertesa però no tenen prou treballada la part de creació de contingut i es fa incòmoda de treballar quan es volen crear elements més complexos que text, taules, imatges i llistes. Afortunadament, un gran percentatge de temes utilitzats per aquests gestors de contingut estan basats en llibreries CSS de popularitat entre la comunitat com *Bootstrap* o *Bulma*, de manera que si existís una eina que permeti generar codi compatible amb *Bootstrap*, aquest codi seria compatible amb les pàgines del gestor de contingut.

ESTUDI DE MERCAT (BENCHMARK)

Abans de començar a desenvolupar el projecte s'ha realitzat una cerca d'eines similars. Les plataformes que ofereixen prestacions afins analitzades són *Bootstrap Studio*, *Android Studio*, *Adobe Dreamweaver*, *Unbounce* i *Wordpress*.

Les eines mencionades són diverses i persegueixen objectius diferents per finalitats diferents, tanmateix, totes tenen una necessitat en comú: crear interfícies gràfiques a través de codi. Pot resultar inspirador veure quines pràctiques segueixen que els doni bons resultats.

BOOTSTRAP STUDIO

Bootstrap Studio és una plataforma per crear webs basades en la llibreria Bootstrap de manera senzilla i intuïtiva.

- **Web:** bootstrapstudio.io
- **Format en què treballa:** Fitxers HTML basats en Bootstrap
- **Cost:** Pagament (\$29 - \$60)
- **Èxit:**
- **Carecterístiques:**
 - Molt complet
 - Hi ha plantilles preparades
 - Interfície estructurada en tres columnes

ANDROID STUDIO

Android Studio és una eina per desenvolupar aplicacions pel sistema operatiu Android, des de la interfície gràfica fins la lògica. Per la comperativa, només ens fixarem en el disseny de la part dedicada a crear les interfícies gràfiques.

- **Web:** developer.android.com
- **Format en què treballa:** Fitxers XML per Android
- **Cost:** Gratuït
- **Popularitat:** Alta
- **Carecterístiques:**
 - Està molt preparat per treballar amb Android
 - Molt complet. Es pot editar cada propietat de cada element
 - Interfície estructurada en tres columnes

ADOBE DREAMWEAVER

Adobe Dreamweaver és un editor de pàgines web WYSIWYG amb suport d'altres tecnologies web com CSS, JavaScript i llenguatges de programació de servidor.

- **Web:** adobe.com
- **Format en què treballa:** Fitxers HTML
- **Cost:** Car (\$20.99/mo)
- **Popularitat:** A la baixa
- **Carecterístiques:**
 - Afegix molt codi redundant
 - Massa car per usuaris ocasionals

UNBONCE

Unbounce és una eina per dissenyar, crear i allotjar *Landing Pages*. Conté una pàgina destinada a crear la part visual de la pàgina amb una interfície gràfica molt complerta.

- **Web:** unbounce.com
- **Format en què treballa:** Landing pages autoallotjades
- **Cost:** Pagament (\$ 3.99-79 USD / mo)
- **Popularitat:**
- **Carecterístiques:**
 - Preparada per usuaris de l'àmbit del màrqueting
 - Afegeix molt codi redundant
 - Interfície estructurada en tres columnes

WORDPRESS

WordPress és un sistema de gestió de continguts (CMS) de codi obert per crear pàgines web. Com que aquesta eina serveix per crear webs complets i la part dedicada a crear estructures és relativament petita i poc treballada.

- **Web:** <https://wordpress.org/>
- **Format en què treballa:** HTML
- **Cost:** Gratuït
- **Popularitat:** Alta
- **Carecterístiques:**
 - Poc complet. No es poden tocar gaires propietats.
 - No hi ha una interfície en tres columnes sinó que es treballa amb finestres emergents. No és gaire pràctic.

Per veure la comparació de carecterístiques, consultin la [taula adjunta als annexos](#).

Després de comparar les diferents aplicacions que hi ha al marcat funcionant, s'han definit els objectius principals del projecte que detallarem al següent apartat.

OBJECTIUS DEL PROJECTE

L'objectiu principal del projecte és aconseguir que els usuaris puguin crear estructures HTML senzilles sense necessitat de conèixer l'estàndard HTML, a més, però, es completarà l'eina amb funcions auxiliars.

Pel que s'ha pogut observar a l'estudi de realitzat en [l'apartat anterior](#), poques eines que permetin crear el codi HTML basar en una llibreria concreta i que sigui fàcil d'utilitzar per a un usuari sense nocions del llenguatge i sense expertesa en el tema.

Aquesta eina està orientada a usuaris que han de crear contingut web però que no són desenvolupadors. No és objectiu del projecte crear o modificar el codi referent als estils

del web, aquesta eina només servirà per crear estructures, ja que normalment, els webs ja tenen un full d'estils existent i preparat en el moment de crear la web.

Per complir aquests objectiu principal, detallarem a continuació un seguit de requisits que haurà de tenir l'aplicació final:

INTERFÍCIE GRÀFICA PER GENERAR CODI

Tota la funcionalitat de l'aplicació es realitzarà bàsicament des d'una mateixa una pantalla del web que de manera senzilla permeti a l'usuari interactuar amb els elements d'una estructura HTML. Aquestes interaccions seran les següents:

- Afegir nous elements al fitxer.
- Eliminar elements del fitxer.
- Modificar les propietats dels elements afegits al fitxer.

A més, aquesta pantalla caldrà que guardi tot el codi generat en fitxers en format html i que permeti emmagatzemar-los al núvol per poder-hi accedir un cop acabat el projecte.

REGISTRE D'USUARIS

Per tal de poder guardar els documents de cada usuari al núvol, cal que els usuaris s'acreditin a l'hora de guardar i visualitzar fitxer, per aquest motiu, cal implementar un sistema d'autenticació d'usuaris.

Una alternativa per evitar l'autenticació d'usuaris seria no guardar fitxers al núvol i que els usuaris descarreguin i carreguin el fitxer cada vegada que l'hagin de modificar. Tanmateix, aquest procés no resulta molt còmode pels usuaris i s'ha optat per obligar a registrar-se al web per poder-hi treballar.

Aquest registre servirà per assignar un usuari als fitxers que hi haurà guardats al servidor i només permetre l'accés a cada fitxer al seu usuari creador.

HISTORIAL DE VERSIONS

Actualment, la majoria d'eines de creació d'elements que permeten guardar dits elements al núvol, inclouen un historial de revisions que permet visualitzar antigues revisions d'un fitxer i recuperar-les si cal.

Per tant, per cada fitxer que hi hagi guardat al servidor, també se'n guardaran versions per tal que l'usuari pugui recuperar els canvis que s'hagin realitzat de manera errònia.

MÚLTIPLES LLIBRERIES CSS

Recordem que el contingut d'una pàgina web està definit en un fitxer HTML i el seu aspecte està definit en un fitxer CSS. Normalment, aquest fitxer CSS està compartit en diferents pàgines i fins i tot existeixen llibreries populars utilitzades per diferents webs.

Per aquest motiu, el generador de codi hauria de ser compatible amb diferents llibreries d'estils i, en mesura del possible, caldria que fos relativament senzill afegir noves llibreries al servidor per facilitar el manteniment del web.

GALERIA MULTIMÈDIA

Sovint, les webs inclouen imatges i altres elements multimèdia per la qual cosa els usuaris tindran la necessitat d'afegir imatges a les seves creacions. Per aquests casos, caldrà allotjar imatges al servidor i per tant desenvolupar una pantalla per tal que cada usuari pugui interactuar amb les imatges que té penjades.

PLANIFICACIÓ

En aquest apartat definirem la planificació temporal que s'ha dut a terme per desenvolupar el projecte:

ETAPES DEL PROJECTE

Les etapes per les quals ha passat el projecte són planificació, desenvolupament, testeig, memòria i millores i idees.

- **Planificació.** L'etapa de planificació és el primer i és l'etapa en què s'ha avaluat la viabilitat del projecte i s'han definit els objectius a complir i s'ha dissenyat el funcionament de l'eina.
- **Desenvolupament.** L'etapa de desenvolupament és la més llarga i important del projecte, ja que és l'etapa en la qual s'han realitzat la major part les funcionalitats de la web. El procés de desenvolupament està documentat més endavant en aquest document.
- **Testeig.** Després de desenvolupar el projecte s'ha dedicat temps a comprovar que el funcionament de la web fos el correcte i s'han arreglat errors de funcionament que s'havien ovbiat durant l'etapa de desenvolupament.
- **Memòria i documentació.** S'ha reservat temps per redactar la memòria del projecte i documentar l'arquitectura d'aquest. Aquesta part no forma part del projecte com a tal però sí que ha requerit esforç.
- **Millores i idees.** Finalment, s'ha deixat un marge temporal al final a tall de pla de contingències que ha acabat sent usat per millorar el projecte i afegir prestacions que estaven pensades per una fase següent.

PLANIFICACIÓ TEMPORAL

La distribució temporal de cada etapa i de les tasques que comporten cada etapa està distribuït en un diagrama de gantt adjuntat als annexos.

Després de planificar el projecte es va desenvolupar l'aplicació web que emmarca l'eina per poder passar a crear una interfície gràfica que de la pàgina que correspon al generador de codi en si. Feta la interfície, s'ha desenvolupat la lògica que comporta. Després s'ha dedicat temps en crear una galeria multimèdia. En aquest punt, el prototip

ja estava presentable, però s'ha invertit temps a preparar el web per tal de ser compatible amb més llibreries CSS, descarregar fitxers i millorar el web.

Moltes de les tasques, estan dividides en franges de dues setmanes per fer-ho encaixar amb les tutories.

ANÀLISI INICIAL DELS COSTOS

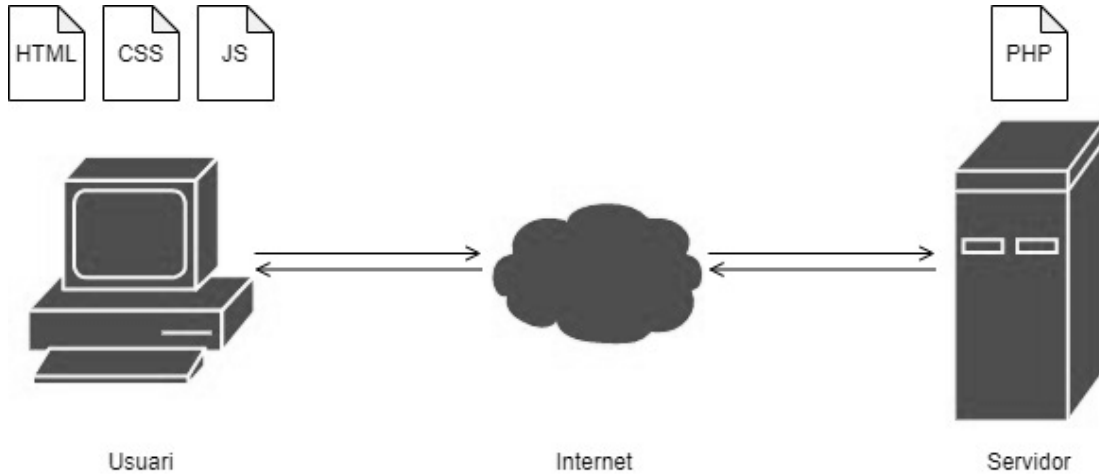
Per tal d'avaluar el cost del desenvolupament del projecte, s'ha realitzat un pressupost

Descripció	Cost		Quantitat	Cost total (€)
Sous (per hora)				
Dissenyador	15.00	€/h	5	75.00 €
Programador	20.00	€/h	80	1,600.00 €
Software i llicències				
Visual Studio Code	-	€/mes	1	- €
XAMPP	-	€/mes	1	- €
Mozilla Firefox	-	€/mes	1	- €
InkScape	-	€/mes	1	- €
Sourcetree	-	€/mes	1	- €
Trello.com	-	€/mes	1	- €
Bitbucket	-	€/mes	1	- €
Frameworks				
Laravel	-	€/mes	1	- €
Vue.js	-	€/mes	1	- €
Sistemes tècnics				
Domini	15.00	€/any	1	15.00 €
Allotjament	120.00	€/any	1	120.00 €
Despeses indirectes				
Electricitat	0.12	€/mes	120	14.40 €
Amortització del PC	15.00	€/mes	3	45.00 €
Subtotal				1,869.40 €

El cost del projecte puja un total de 1869,40€. Pel que fa el desenvolupament i el primer any de funcionament. Després tindrà 135€ de despesa anuals.

TECNOLOGIES UTILITZADES

Una aplicació web està desenvolupada mitjançant diferents llenguatges de programació. Aquests llenguatges són HTML (per definir l'estructura dels continguts), CSS (per definir els estils i estètica dels continguts) i PHP (per programar la lògica del servidor).



IL·LUSTRACIÓ 3 ESQUEMA DE L'ARQUITECTURA CLIENT SERVIDOR

A més, una web pot necessitar executar programes en la màquina de l'usuari. Aquests programes han de ser en llenguatge JavaScript que és el llenguatge de programació que poden interpretar i executar els navegadors.

FRAMEWORKS DE DESENVOLUPAMENT

Per desenvolupar la part de servidor que generarà les pantalles i interfícies que s'enviaran al client del projecte s'ha utilitzat **Laravel**. Laravel és un framework de desenvolupament PHP de codi lliure pensat per construir aplicacions web seguint l'arquitectura model-vista controlador (veure més informació a l'apartat [Arquitectura](#)). Utilitzar aquesta eina suposa un estalvi temporal significatiu i una sintaxi més clara i entenedora als fitxers de codi ja que és molt modular. S'ha optat per Laravel perquè ja es disposava d'experiència prèvia i, a més, és el framework client-servidor en llenguatge més àmpliament utilitzat.

La interfície d'usuari s'ha desenvolupat en **Vuejs**, que comporta una arquitectura incrementalment adaptable que se centra a renderitzar components prèviament declarats. A més, és una arquitectura reactiva, de manera que el codi reacciona i actualitza les dades cada vegada que l'usuari interacciona amb la interfície. S'ha optat per aquesta eina perquè s'integra molt bé amb l'arquitectura que proposa Laravel i, de fet, ja ve integrada en el mateix framework.

SOFTWARE

Per desenvolupar el projecte s'han hagut d'utilitzar altres *softwares*, tanmateix, aquests han estat escollits per comoditat però no han estat de vital importància pel projecte:

- **Visual Studio Code** (editor de text) per construir el codi de la web.

- **XAMPP** (Apache i MySQL) per emular un servidor web en local.
- **Mozilla Firefox** (navegador) per provar i executar el projecte.
- **InkScape** (eina de dibuix vectorial lliure) per crear el logotip del web i fer alguns dissenys.
- **Sourcetreer + Bitbucket** (control de versions) per guardar un historial de versions del projecte.
- **Microsoft Office** (editor de textos, full de càlcul i presentacions de diapositives) per crear la memòria del projecte i la seva presentació.

DESENVOLUPAMENT DEL PROJECTE

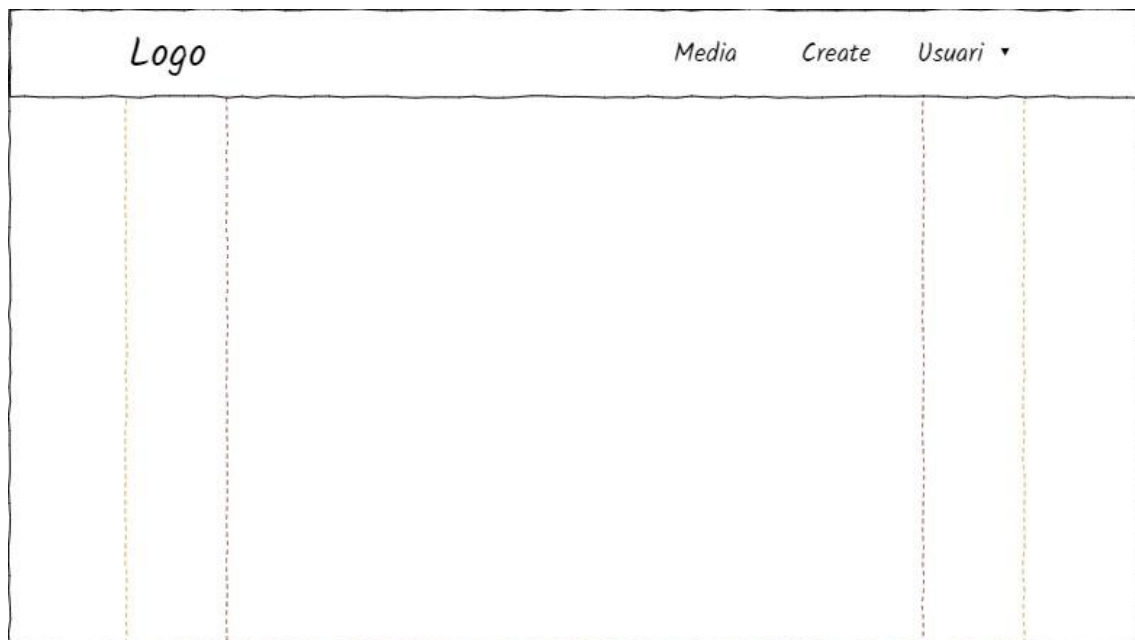
El desenvolupament del projecte s'ha dividit en diferents parts on cada part engloba diferents tasques. Cada part s'ha desenvolupat en franges de dues setmanes i consecutivament mentre que les tasques dins de cada part s'han desenvolupat de manera més àgil.

DISSENY

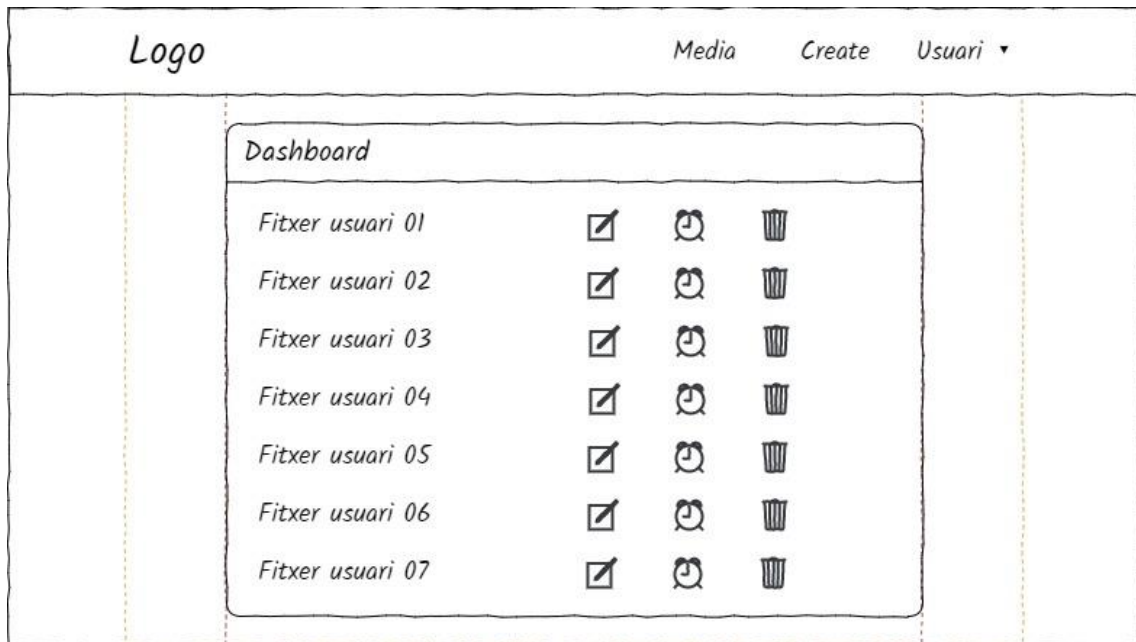
Abans de començar a desenvolupar cal planificar gràficament com serà l'aplicació i definir l'estètica del projecte.

User experience

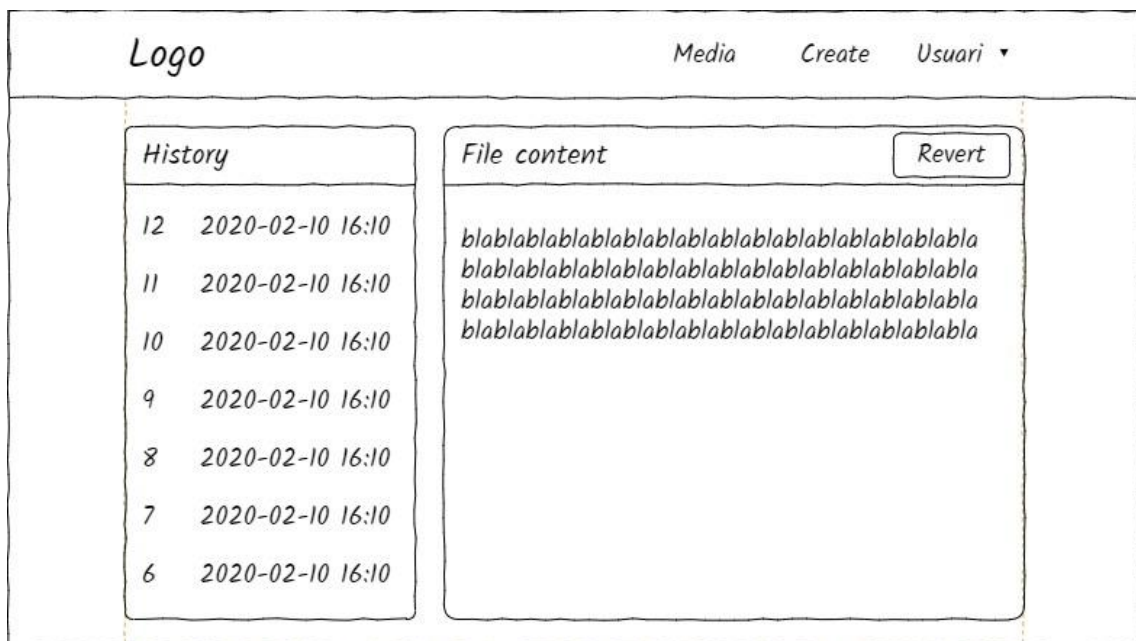
Per partir d'una base a l'hora de començar a desenvolupar l'aplicació s'ha dissenyat la distribució d'elements dins de cada pantalla.



IL·LUSTRACIÓ 4 ESQUEMA DE LA DISTRIBUCIÓ BÀSICA DE L'APLICACIÓ



IL·LUSTRACIÓ 7 ESQUEMA DE LA DISTRIBUCIÓ DE LA PÀGINA DE TAULELL D'USUARI



IL·LUSTRACIÓ 8 ESQUEMA DE LA DISTRIBUCIÓ DE LA PÀGINA D'HISTORIAL

LOGO

S'ha elaborat un imatgotip per donar professionalitat a la imatge del projecte.



IL·LUSTRACIÓ 9 IMAGOTIP DEL WEB

LLIBRE D'ESTIL

L'estètica del projecte s'ajusta als estils que Laravel porta predefinitos. D'aquesta manera l'aplicació té una aparença elegant sense la necessitat d'una gran inversió temporal.

Alguns paràmetres de l'estètica s'han modificat per personalitzar una mica el web. Per exemple, la tipografia del web és *Nunito* que presenta un aspecte formal i modern. S'ha invertit l'escala de grisos aconseguir un tema fosc i en conseqüència donar una aparença més moderna. També s'ha definit el color magenta i taronja com a colors principal i secundari.

#c50957	#e27a0b
Color Principal	Color Secundari

APLICACIÓ WEB BÀSICA

El primer pas és crear la plataforma base de la web i una estructura mínima. Per facilitar el desenvolupament, s'ha optat per utilitzar el *framework* de desenvolupament *Laravel* i no partir des de zero.

1. Una de les prestacions que *Laravel* ja porta semi-implementada és la gestió d'usuaris de manera que en aquest projecte s'ha hagut de configurar aquesta prestació però no s'ha implementat íntegrament.
2. Com que la web ha de constar de diferents pàgines, s'han preparat els fitxers pel model-vista-controlador i s'han afegit els enllaços al *layout* base.
3. S'ha creat i implementat una base de dades per guardar tota la informació dels usuaris i els fitxers de l'aplicació.
4. S'ha preparat l'arquitectura per comunicar l'aplicació amb la base de dades.

CRUD DE FITXERS I REVISIONS

Un cop obtinguda l'aplicació web bàsica, cal preparar l'arquitectura per la qual el web permet Crear, Editar i Eliminar fitxers.

1. Disseny i arquitectura de la vista del tauler d'usuari. Aquesta vista ha de contenir una llista amb tots els fitxers i per cada fitxer hi ha d'anar un enllaç per editar el fitxer, veure l'historial de revisions i eliminar el fitxer.

2. Disseny i arquitectura d'una vista per previsualitzar el fitxer (més endavant, serà la pantalla per generar codi, però de moment, només mostra el codi del fitxer).
3. Formulari per crear un fitxer nou.
4. Preparar la lògica necessària per editar fitxers i guardar els canvis.
5. Guardar una revisió nova cada vegada que es guarda un canvi nou en un fitxer.
6. Preparar la lògica necessària per eliminar el fitxer.

EMMAGATZEMATGE DE COMPONENTS

La informació dels components HTML de cada llibreria han d'estar guardats en alguna base de dades. Per tal de poder afegir noves llibreries amb facilitat, s'ha optat per guardar aquesta informació en documents JSON. Això permetrà que en un futur el projecte sigui més fàcilment ampliable sense requerir, en molts casos, modificacions en el codi.

1. Dissenyar un *JSON* amb tota la informació necessària
2. Crear un *JSON* amb components bàsics
3. Implementar una arquitectura que llegeixi *JSONs*

ESTRUCTURA BÀSICA CLIENT DE LA PÀGINA STUDIO

La pàgina Studio serà la pàgina principal del projecte que contindrà l'eina del generadora de codi.

Crear una primera versió de la pantalla `/studio` que contingui una columna amb els components disponibles, una amb la previsualització del que s'està creant i una tercera amb les propietats de l'element seleccionat.

El programa referent a aquesta part ha d'executar-se al navegador del client, de manera que caldrà que estigui implementat en *Javascript*. Per desenvolupar aquesta part, s'ha optat per utilitzar *vuejs*.

1. Preparar la interfície en tres columnes
2. Fer el component bàsic per la columna "components"
3. Fer el component bàsic per la columna "codi"
4. Modificar el codi HTML resultant quan es clica un component
5. Fer el component bàsic per la columna "propietats"
6. Modificar el codi quan s'editen les propietats
7. Fer una interfície dinàmica per la columna "propietats"

ACCESSIBILITAT CLIENT DE LA PÀGINA STUDIO

Programar la pàgina `/studio` per tal que els usuaris puguin editar els seus documents amb *drag and drop* de manera còmode i intuïtiva.

1. Renderitzar la vista i mostrar una previsualització millor del codi
2. Implementar una navegació *drag and drop* per afegir elements
3. Clicar un element per editar-lo
4. Afegir un botó per eliminar elements

ELEMENTS DINS D'ELEMENTS

Aconseguir guardar *elements fills* als components aplicats per l'usuari a `/studio`. Això implica crear algorismes recursius per llegir i renderitzar els fitxers.

- Crear l'esdeveniment que afegixi fills a l'estructura abstracta.
- Modificar els algorismes de processament del codi HTML per tal que siguin recursius.
- Fer que determinats elements només permetin certs fills.
- Fer que certs elements només puguin ser fills de certs pares.

PENJAR IMATGES

Si els usuaris necessiten afegir imatges als fitxers, necessitaran una opció que els permeti penjar imatges al servidor.

1. Crear una pantalla de per gestionar les imatges.
2. Formulari de càrrega d'imatges.
3. Opció per eliminar imatge de la galeria.
4. Clicar per veure els detalls de la imatge a `/media`.
5. Vincular la galeria a la informació del component imatge a `/studio`.

MÚLTIPLES LLIBRERIES CSS

Afegir noves llibreries diferents de *bootstrap*.

1. Crear un menú per seleccionar la llibreria en qüestió.
2. Aplicar estils propis de la llibreria a la pantalla *preview*.
3. Crear nous *JSON* amb noves llibreries.

OPCIÓ DE DESCÀRREGA

Els usuaris han de tenir l'opció de descarregar els fitxers que creïn. Per aquesta finalitat, cal implementar un botó que generi tot el codi en HTML net i el descarregui a l'usuari.

1. Preparar l'arquitectura del botó de descarregar.
2. Renderitzar el codi a HTML.

HISTORIAL DE REVISIONS

Finalment, cal fer un historial de revisions que permeti visualitzar i revertir les revisions existents.

1. Crear la pantalla de revisions.
2. Carregar la llista de revisions del fitxer.
3. Previsualitzar revisions.
4. Maquetar l'historial.

ARQUITECTURA DE L'APLICACIÓ

A continuació parlarem del funcionament intern de l'aplicació i les seves respectives capes.

SERVIDOR (LARAVEL)

L'arquitectura de l'aplicació segueix l'esquema de *Model, Vista i Controlador* (MVC). On el *model* és la capa que gestiona les dades i les connexions amb la base de dades. La *vista* és la presentació visual que rebrà l'usuari i el controlador és la capa que controlarà la lògica de l'aplicació.

Laravel ja porta implementades les capes necessàries per treballar en MVC. A continuació es comenten els fitxers que hi ha a cada capa:

MODELS

- **File** (fitxer): els fitxers són els documents que creen els usuaris. Per cada fitxer es guarda el nom, l'usuari que n'és el propietari i la llibreria en què està basada.
- **Revision** (revisió): les revisions són les versions que ens van guardant cada vegada que es guarda cada vegada que l'usuari clica el botó de guardar. Per cada revisió es guarda el fitxer a què pertany, el número de revisió i la data.
- **User** (usuari): els usuaris són les persones que tenen una compta guardada a l'aplicació. Per cada usuari es guarda el correu electrònic, el hash de la contrasenya i el nom d'usuari.

VISTES

Les vistes i pantalles que té l'aplicació són:

- **Tauler d'usuari**: aquí l'usuari tindrà una llista amb tots els seus fitxers i la possibilitat d'editar-los, eliminar-los o accedir al seu historial.
- **Historial de revisions**: és la pantalla que mostra la llista de revisions que té cada fitxer.
- **Portada**: és una pantalla estàtica on els usuaris poden informar-se sobre l'aplicació.
- **Media**: és la pantalla on es poden penjar i visualitzar les imatges al servidor.
- **New**: aquest és un petit formulari per crear un fitxer nou.
- **Studio**: aquesta serà la pantalla més complexa, ja que serà l'eina principal que permetrà crear les estructures web.

Cada vista té el seu controlador associat on es programarà la lògica de les diverses accions que permet realitzar la vista.

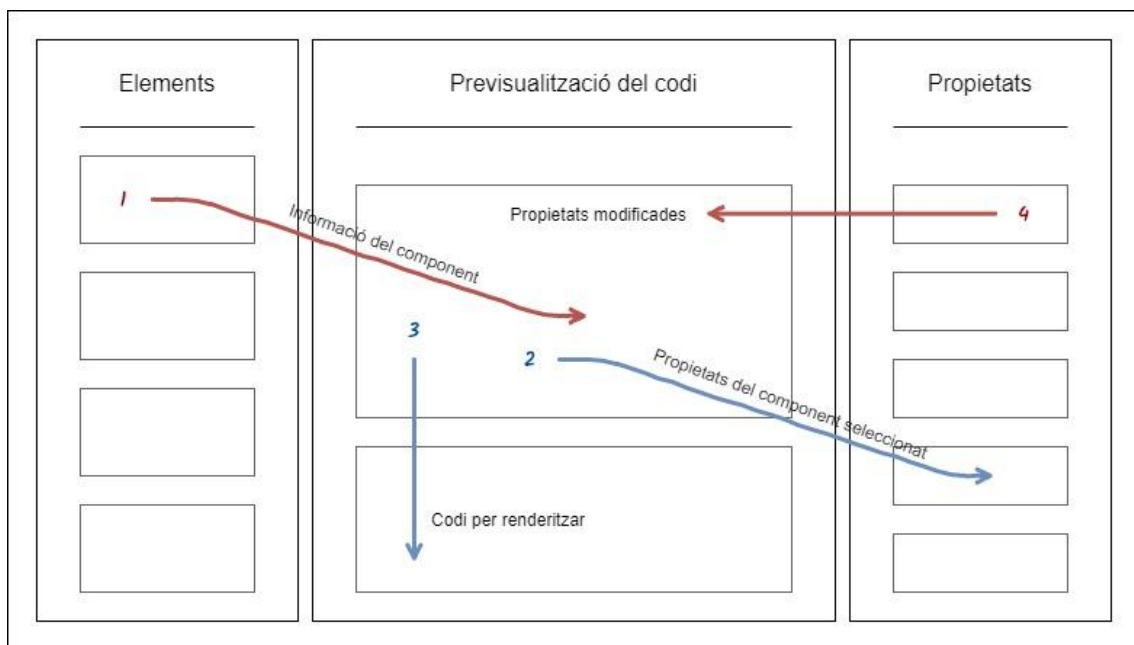
CLIENT (VUE.JS)

La pantalla *studio* s'executa a la banda client per evitar saturar el servidor i millorar la velocitat. Com s'ha comentat, la programació de les vistes que requereixen executar codi a la part de client s'ha implementat amb vuejs que és una llibreria de javascript.

Vuejs funciona amb components. Cada component és com una petita aplicació que fa el seu codi independentment dels altres components. Per la pantalla *studio* s'han creat 4 components.

El component principal és la pantalla mateixa i conté als altres tres components que són:

- La columna d'elements
- La previsualització del codi
- La columna de propietats



IL·LUSTRACIÓ 10 ESQUEMA DELS COMPONENTS DE VUEJS

La **columna d'elements** és la columna que conté una llista amb tots els elements que es poden arrossegar fins al codi. Aquesta columna llegeix la llista d'elements del document JSON i comprova si aquests elements són compatibles amb l'element seleccionat. En cas que l'usuari comenci a arrossegar un element cap al codi, aquest component envia les dades necessàries al següent component.

La **columna de previsualització de codi** és la columna que l'usuari utilitzarà per interactuar amb el codi. La feina d'aquest component és rebre el codi html del component principal i renderitzar-lo, a més, també ha de rebre els components que l'usuari deixi anar a sobre i enviar la informació al component principal.

L'última columna, la de **propietats**, ha de mostrar les propietats del component que hi hagi seleccionat per l'usuari juntament amb un formulari per modificar-les. Òbviament, si es modifica una propietat, cal comunicar-ho al component principal.

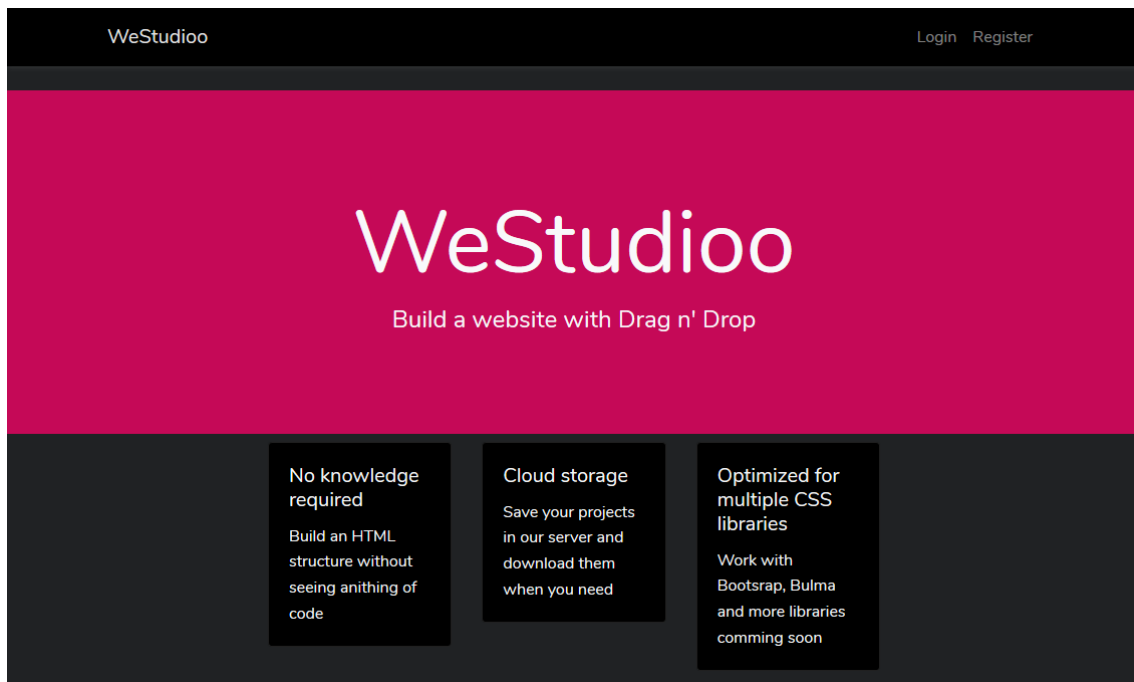
El component principal és qui gestiona l'estructura de dades del document i es comunica amb els altres components.

RESULTATS

Els resultats aconseguits, tal com s'havia definit en un inici inclouen una gestió d'usuaris, un historial de revisions, un tauler d'usuari, una interfície d'usuari còmode i també s'ha dissenyat una portada comercial.

PORTADA

Els usuaris que arribin al web es trobaran una pàgina que no té cap utilitat a part de informar:



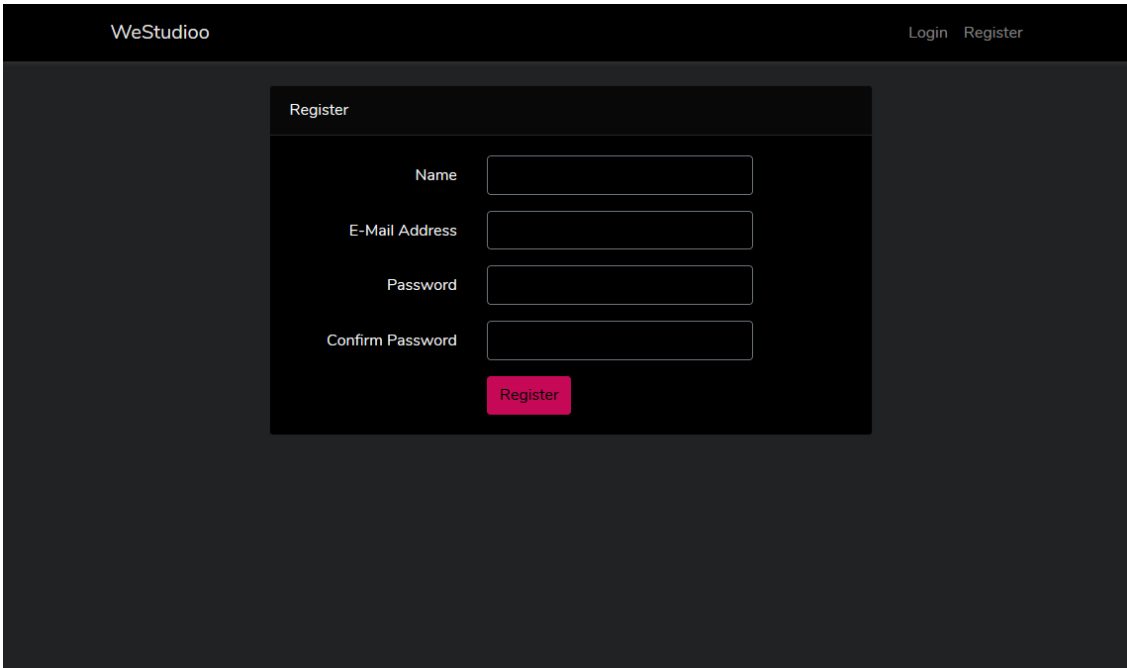
IL·LUSTRACIÓ 11 PORTADA DEL WEB (/)

GESTIÓ D'USUARIS

Com a usuari, el primer pas per començar a utilitzar l'aplicació serà crear un usuari. L'usuari podrà registrar un usuari anant a `/register` o bé clicant l'enllaç 'Login' de la barra superior.

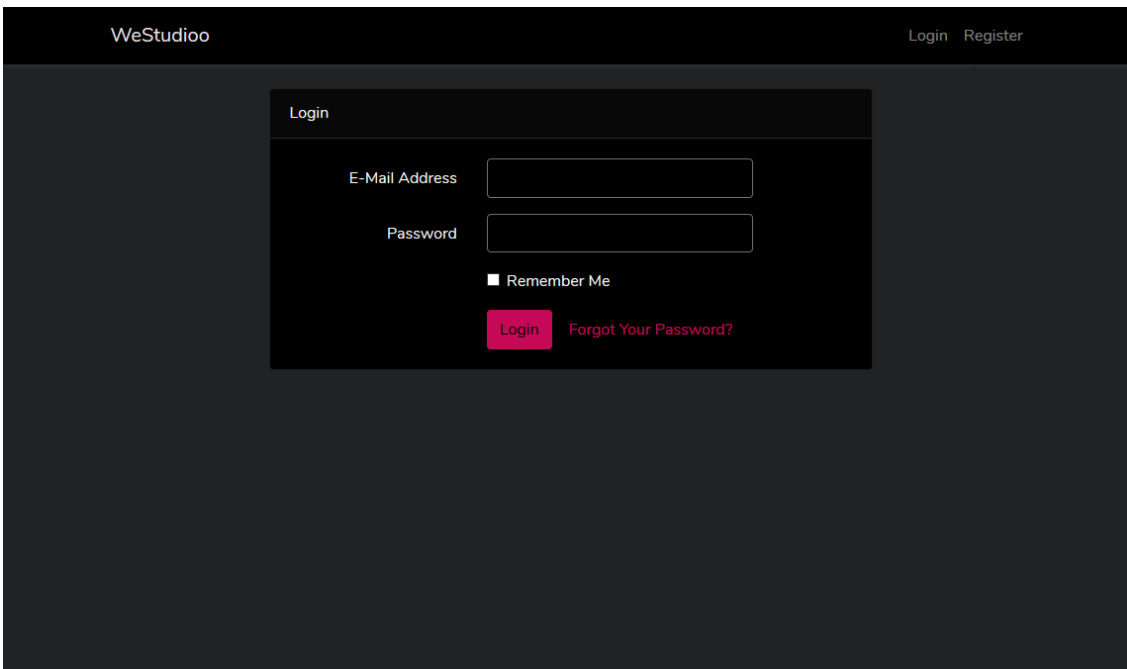
Un cop a dins la pantalla de registre caldrà que l'usuari empleni el formulari amb les credencials que li interessen. Si hi ha algun problema de l'estil que el correu electrònic ja estigui en ús o que les contrasenyes no coincideixin, el programa ho notificarà a l'usuari.

Actualment, el web no està en fase de producció, però quan hi estigui, caldrà confirmar el correu electrònic mitjançant un enllaç.



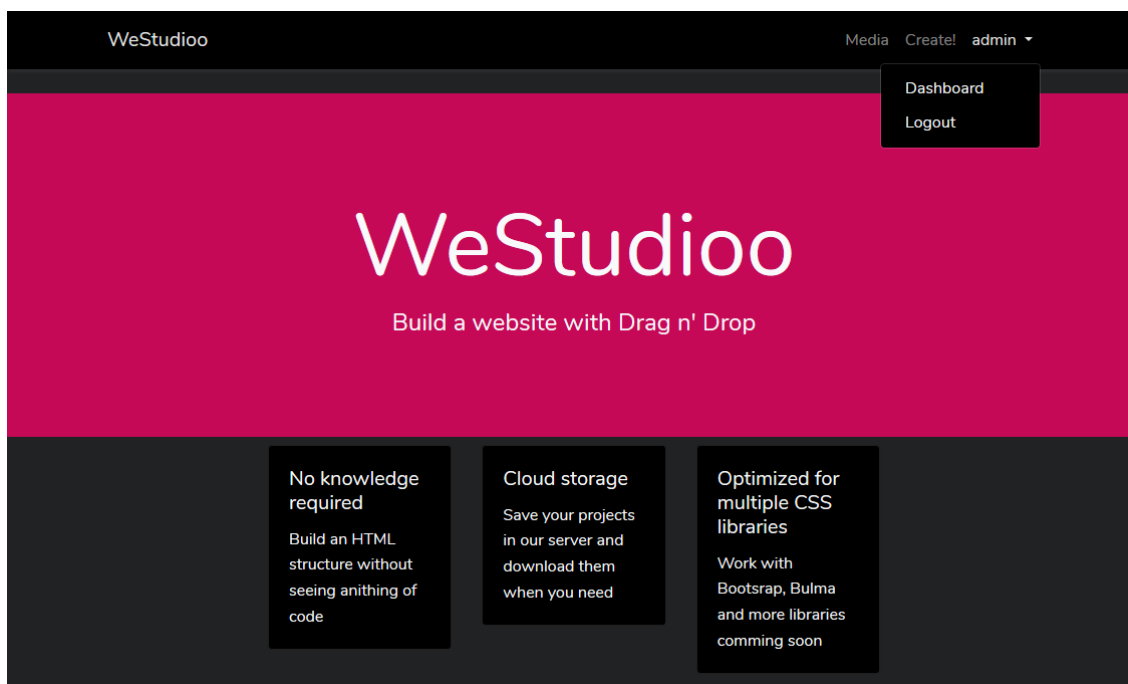
IL·LUSTRACIÓ 12 PÀGINA DE REGISTRE D'USUARIS (/REGISTER)

En cas de tenir un usuari creat prèviament, l'usuari accedirà a `/login` i entrarà les credencials que es van definir quan es va crear l'usuari per autenticar-se.



IL·LUSTRACIÓ 13 PÀGINA D'AUTENTICACIÓ D'USUARIS (/LOGIN)

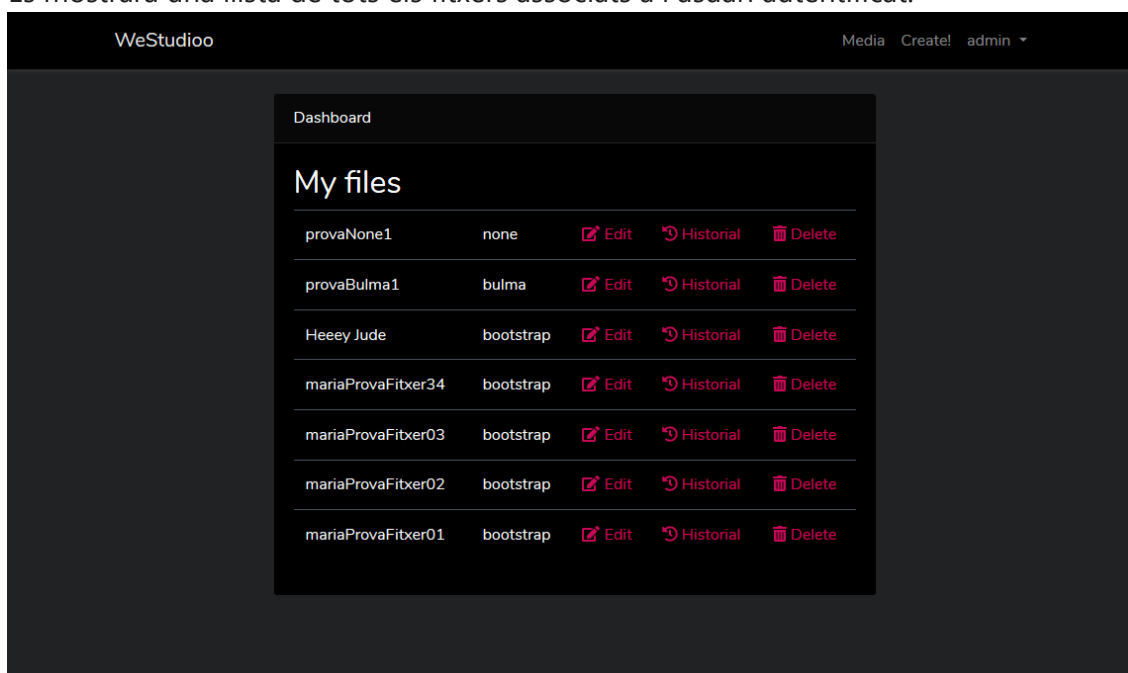
Un cop l'usuari s'hagi autenticat, els enllaços de la barra superior canvien als següents:



IL·LUSTRACIÓ 14 PORTADA DESPRÉS D'AUTENTIFICAR L'USUARI (/)

TAULER USUARI

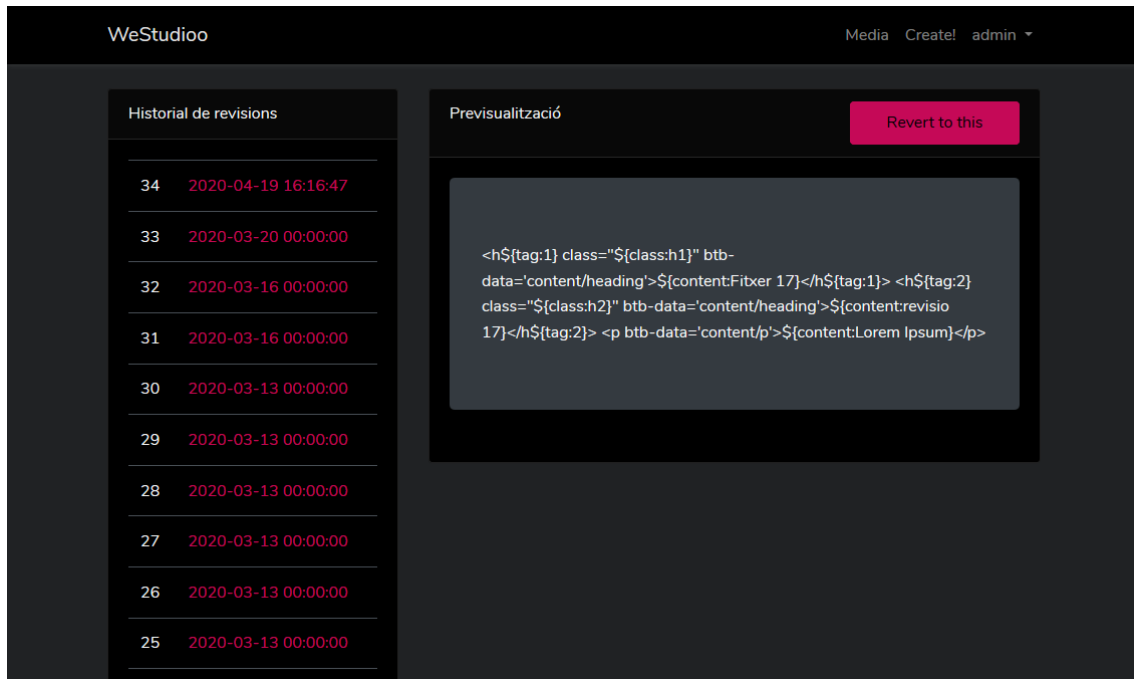
Cada usuari disposarà d'un espai privat amb els seus fitxers. S'hi accedeix a /dashboard. Es mostrarà una llista de tots els fitxers associats a l'usuari autenticat.



IL·LUSTRACIÓ 15 TAULER DE L'USUARI (/DASHBOARD)

Al costat de cada fitxer, hi ha l'opció per accedir a l'historial de revisions de l'esmentat fitxer.

Aquesta pantalla mostra el llistat de revisions de l'usuari. Cada revisió és clicable i permet visualitzar el text que correspon a cada revisió.

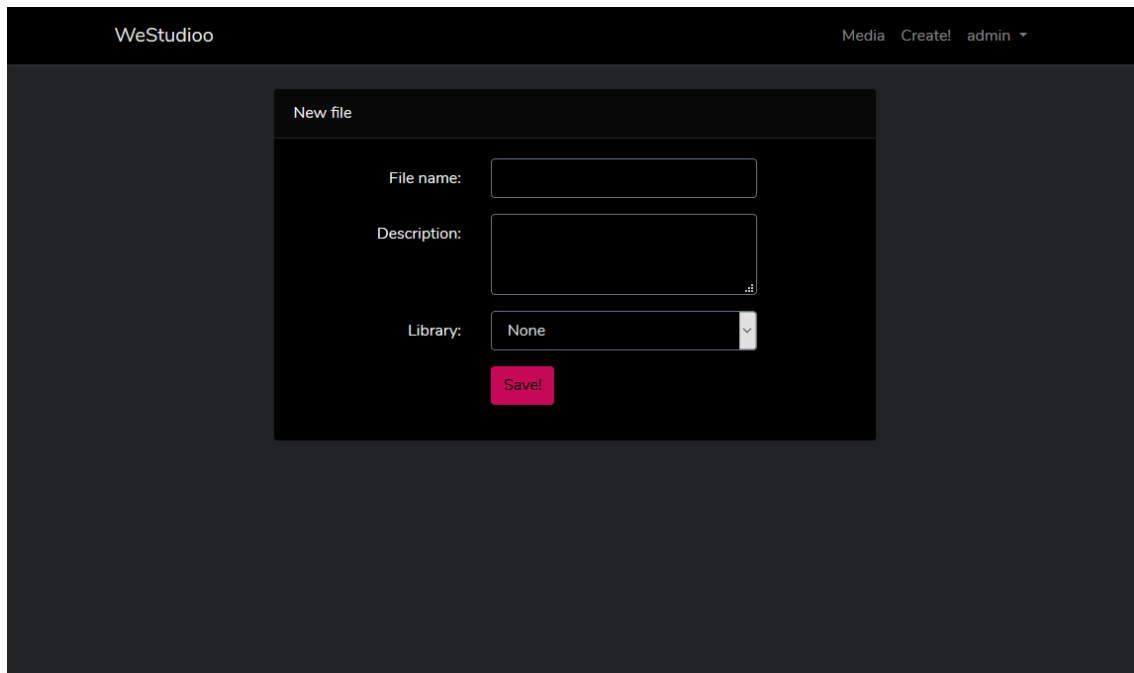


IL·LUSTRACIÓ 16 HISTORIAL DE REVISIONS D'UN FITXER (/HISTORY)

Al tauler, també apareix l'opció que conduirà l'usuari a la pantalla *Studio*.

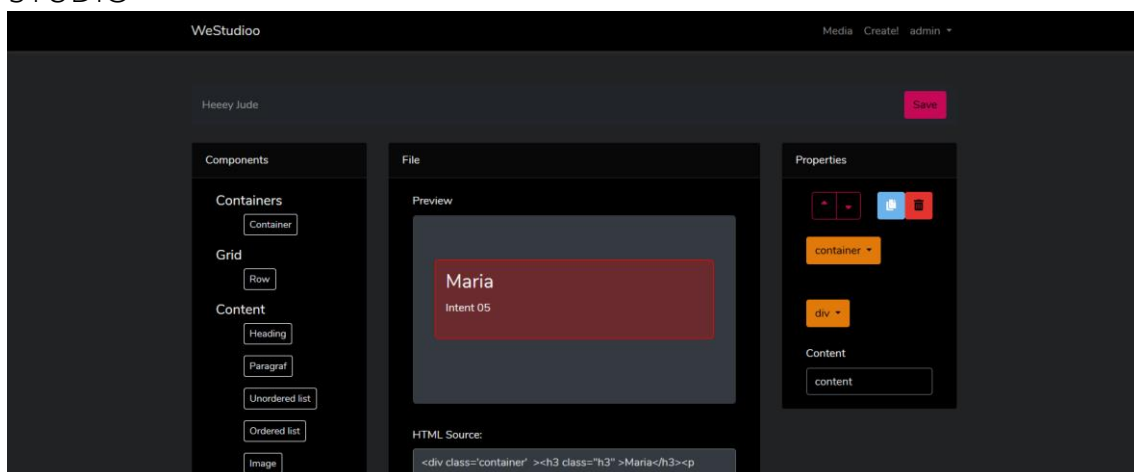
CREATE

Per crear un fitxer l'usuari aniria a /create o bé clicaria l'enllaç 'Login' de la barra superior. Aquesta pantalla mostra un formulari on es demanen les dades bàsiques per crear un fitxer nou. Quan l'usuari clica el botó "Save!" es crea el fitxer amb una primera versió i es reconduïx a l'usuari a la pantalla *Studio*.



IL·LUSTRACIÓ 17 FORMULARI PER CREAR UN FITXER (/CREATE)

STUDIO



IL·LUSTRACIÓ 18 PANTALLA STUDIO (/STUDIO)

La pantalla d'estudi està formada per tres columnes. La primera columna hi ha el llistat de components que es poden afegir al codi, la segona conté la previsualització del codi i la tercera conté les propietats del component seleccionat.

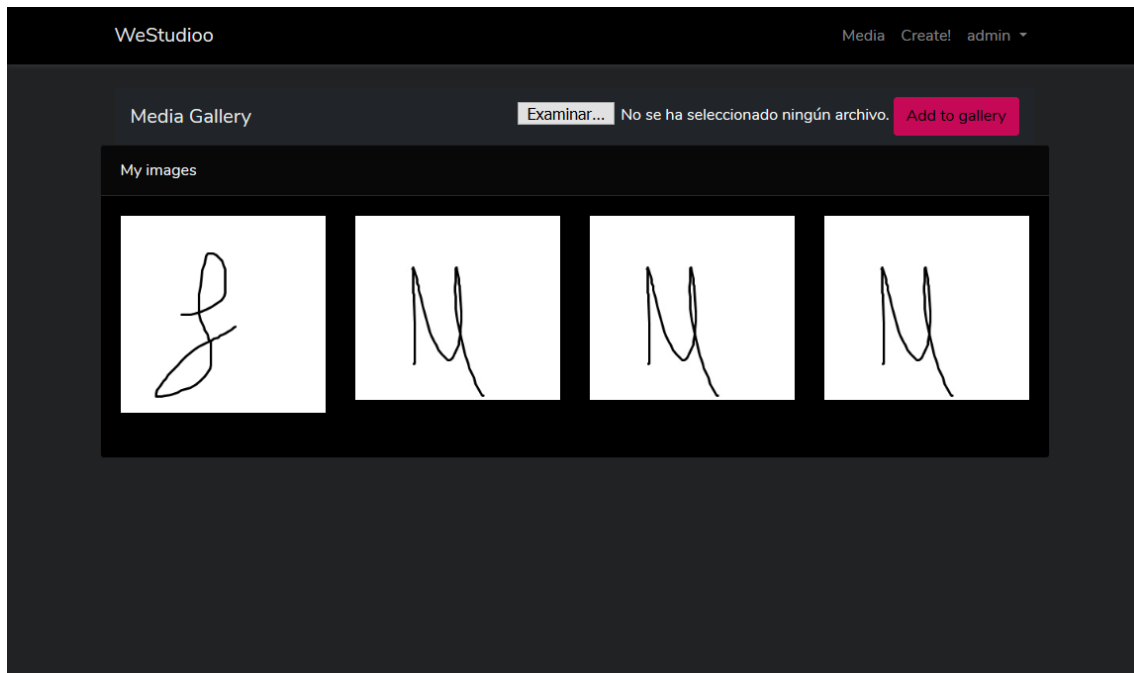
Per afegir un element nou, l'usuari ha d'arrossegar el component desitjat des de la primera columna cap a la segona. Podem deixar anar el nou component al final del codi o dins d'un altre element. Cal tenir en compte que no tots els elements poden estar dins de qualsevol element.

Un cop s'hagi afegit un element nou al fitxer, aquest queda seleccionat. Per seleccionar un altre element, caldria clicar sobre l'element des de la columna de codi.

Un cop tenim un element seleccionat el podem editar-ne les propietats des de la tercera persona. A més també es poden realitzar les següents accions:

- **Moure:** amb les fletxes que apareixen a la part superior de la columna
- **Eliminar:** amb el botó vermell que apareix a la part superior de la columna
- **Duplicar:** el botó blau que apareix a la part superior de la columna

MEDIA



IL·LUSTRACIÓ 19 PANTALLA DE LA GALERIA MULTIMÈDIA (/MEDIA)

La pantalla *media* permet que els usuaris puguin penjar imatges a la plataforma per poder crear fitxers de codi. També permet visualitzar les imatges que l'usuari ja ha penjat prèviament i consultar les propietats bàsiques de cada imatge.

DISCUSSIÓ I TREBALL FUTUR

Un cop donat el prototip del projecte per acabat els resultats són satisfactoris, ja que s'han complert els objectius proposats. Tanmateix, el projecte no està acabat i queda encara molt camí per recórrer.

Un dels objectius inicials era crear una galeria multimèdia i s'ha complert però només permet penjar imatges mentre que en una fase més avançada hauria de suportar formats de vídeo i àudio. Un altre objectiu era l'historial de revisions. Aquest s'ha creat satisfactòriament però de moment no suporta previsualitzar el codi de cada fitxer.

Pel que fa al generador de codi i partit principal del projecte, s'ha aconseguit prototipar l'eina però caldria polir el funcionament per fer-lo més amigable per l'usuari. Per exemple, s'hauria de permetre arrossegar elements cap amunt o cap avall un cop

col·locats al llenç. A més, hi ha poques propietats personalitzables a la columna de propietats.

Òbviament, també caldria afegir compatibilitat amb més llibreries CSS, ja que de moment només n'hi ha dues. Tanmateix, tal com s'ha desenvolupat el web aquesta tasca no requereix una gran inversió temporal.

Finalment, cal mencionar que s'han deixat algunes idees per a futures fases del projecte:

- Implementar un **autoguardat automàtic** al generador de codi de manera que els usuaris no hagin d'estar preocupats.
- Ressaltar la **la sintaxi del codi HTML** mentre es va generant. Potser hi ha usuaris que volen consultar el codi i els resultaria més fàcil poder xafardejar el codi.
- **Publicar web** a un servidor per fer-la pública i visible per tothom.
- Permetre **importar fitxers HTML** i editar-lo des de l'aplicació web.
- **Importar llibreries CSS des d'un JSON**. Aquesta opció permetria als usuaris poder treballar amb els seus propis fulls d'estils!
- **Documents col·laboratius**. Seria pràctic que els usuaris poguessin compartir documents entre ells.

Malgrat les evidents millores que el projecte té, els resultats obtinguts són els esperats tenint en compte les possibilitats inicials.

CONCLUSIONS

Durant el grau i sobretot durant els projectes que se'ns encomanava desenvolupar des de cada assignatura he necessitat realitzar cerques per internet per resoldre problemes puntuals i per estendre el coneixement adquirit a les aules. En el món del desenvolupament hi ha una gran comunitat subministrant coneixement i eines de manera altruista i com a desenvolupadora hi volia col·laborar.

Des del punt de vista acadèmic, aquest projecte m'ha obligat a utilitzar una gran part de les habilitats adquirides durant el grau, des de disseny fins a programació passant per màrqueting i l'experiència d'usuari.

El repte més gran ha estat dissenyar els algorismes necessaris per convertir l'estructura de dades del fitxer d'un usuari a una estructura HTML i viceversa, ja que hi ha múltiples casuístiques que cal haver previst.

Aquest projecte ha resultat ser enriquidor i motiva a seguir treballant en l'àmbit del desenvolupament d'aplicacions en el futur.

REFERENCES

Bootstrap. <https://getbootstrap.com/>. [En línia]

Google Fonts. <https://fonts.google.com/>. [En línia]

Laravel Documentation. <https://laravel.com/docs/7.x>. [En línia]

Vuejs. <https://vuejs.org/>. [En línia]

ANNEXOS

BENCKMARK: TAULA COMPRARATIVA

	Bootstrap Studio	Android Studio	Adobe Dreamweaver	Unbounce	Wordpress
Cost	Green	Green	Red	Red	Green
Intuitiu	Yellow	Orange	Orange	Yellow	Orange
Multiplataforma	Yellow	Orange	Orange	Green	Green
Emmagatzematge al núvol	Orange	Red	Orange	Green	Green
Múltiples llibreries	Red	Red	Orange	Red	Yellow
Capacitat de personalització element per element	Green	Green	Green	Green	Orange
Exportació de codi	Green	Green	Green	Red	Yellow
Edició per codi	Red	Green	Green	Yellow	Orange
Previsualització WYSIWYG	Green	Green	Green	Green	Green

PLANIFICACIÓ TEMPORAL: DIAGRAMA DE GANTT

