



Realització d'un CMS estàtic

a partir de la web HTML ja existent

wefer.com web ferroviaria

Memòria de Projecte Final de **Grau Multimèdia**

Gestió i publicació de continguts

Autor: **Marçal Guardiola Sánchez**

Professor: **Joan Giner Miguelez**

Lliurament final - Juny de 2020



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

« El minimalisme no consisteix en tenir menys del que necessites. Es tracta de posseir exactament allò que necessites »

Joshua Becker

« Quant més tens, més ocupat estàs. Quant menys tens, més lliure ets »

Mare Teresa

Abstract

L'objectiu d'aquest treball és la remodelació d'una antiga pàgina web dedicada a la divulgació històrica i tècnica del ferrocarril a Espanya, que data de 1997, i que presenta una gran quantitat d'informació i mitjans gràfics útils.

Aquesta web utilitza tecnologies obsoletes que dificulten tant la lectura en dispositius moderns com la indexació en els motors de cerca actuals.

Per a la seva realització s'ha optat per utilitzar el programari Publii, en fase beta, que permet crear pàgines estàtiques en HTML+CSS visualment (WYSIWYG) de manera similar a com es treballa amb un CMS convencional.

A més s'aprofita per comparar el sistema de treball amb el d'altres generadors de pàgines web estàtiques actuals.

L'actual projecte és únicament la presentació de l'estructura de wefer.com amb exemples de cada apartat, ja que el lloc web definitiu necessitarà de mesos/anys per arribar a estar complet en quant a contingut. En la versió actual únicament s'inclouen 105 pàgines i 2.000 fotografies, representant menys d'un 20% del resultat final a llarg termini.

Lloc web de prova per visualitzar l'estat de l'aplicació:

<https://www.wefer.com>

Índex general

Abstract	4
Índex general	5
Índex annexos	6
Figures i taules	7
Canvis des de la versió 3 a la versió final de la memòria.....	9
1. Introducció	10
2. Descripció	12
3. Objectius	14
3.1. Objectius principals	14
3.2. Objectius secundaris	14
4. Marc teòric	15
5. Continguts.....	17
6. Metodologia	19
7. Arquitectura de l'aplicació	20
8. Plataforma de desenvolupament	22
9. Planificació	23
10. Procés de desenvolupament	25
11. APIs utilitzades	27
12. Prototips	29
13. Perfils d'usuari	31
14. Usabilitat i experiència d'usuari	32
15. Seguretat	33
16. Tests	34
17. Versions de l'aplicació	35
18. Requisits d'instal·lació	36
19. Instruccions d'instal·lació	38
19.1. Creació del lloc web	38
19.2. Configuració del servidor	38
19.3. Personalització: temes	39
20. Instruccions d'ús (edició)	44
20.1. Creació d'articles	44
20.2. Inclusió d'imatges	48
20.3. Altres funcions	50
21. Errors ("bugs")	53
21.1 Backoffice: Publii	53
21.2 Frontoffice. Pàgina publicada	54

22.	Pressupost	55
23.	Anàlisi de mercat	56
23.1	Sistemes per a la realització de webs estàtics	56
23.2.	Anàlisi d'un alternativa: Jekyll	57
23.2.1.	Què és Jekyll?	57
23.2.2.	Instal·lació de Jekyll	57
23.2.3.	Funcionament de Jekyll	59
23.2.4.	Markdown	61
23.2.5.	Jekyll vs. Publii	64
24.	Conclusions	66
24.1.	Avantatges i desavantatges dels generadors de llocs web estàtics	66
24.2.	Procés de creació amb Publii	67
24.3.	Comparació Publii-Jekyll	69
25.	Vita Marçal Guardiola Sánchez	71

Índex annexos

Annex 1.	Lliurables del projecte	72
Annex 2.	Codi font (extractes)	73
A2.1	- HTML de la pàgina inicial	73
A2.2	- HTML de les pàgines d'estacions	75
A2.3	- HTML de les pàgines d'unitats	77
A2.4	- Estils CSS	79
Annex 3.	Llibreries/Codi extern utilitzat	83
Annex 4.	Captures de pantalla	84
Annex 5.	Llibre d'estil	104
Annex 6.	Glossari	106
Annex 7.	Bibliografia	108

Figures i taules

- *Fig. 01: aspecte actual de la pàgina inicial, realitzada amb taules i "frames"*
- *Fig. 02: aspecte actual de la pàgina d'estacions a remodelar*
- *Fig. 03: interfície de l'aplicació Publii a Windows 10*
- *Fig. 04: comparació de velocitat entre HTML i el llenguatge dinàmic JSP (Java Server Pages). Font: "A Web Coding blog"*
- *Fig. 05: estructura de wefer.com*
- *Fig. 06: funcionament esquemàtic d'una petició de pàgina web per part d'un client*
- *Fig. 07: PLESK, un dels gestors que ens permet controlar els paràmetres del nostre lloc web*
- *Fig. 08: funcionament d'un servidor DNS per a la resolució de noms de domini a internet. Font: kaspersky.es*
- *Fig. 09: temporització del projecte (diagrama de Gantt)*
- *Fig. 10: finalització el procés de sincronització*
- *Fig. 11: edició HTML amb dremweaver*
- *Fig. 12: editor CSS intern de Publii*
- *Fig. 13: prototip fet en HTML+CSS*
- *Fig. 14: un dels primers prototips fent servir la plantilla Square*
- *Fig. 15: aspecte pròxim al definitiu de la pàgina inicial*
- *Fig. 16: exemple de la cerca "trens" a Google imatges, una de les possibles entrades d'usuaris al lloc web*
- *Fig. 17: test d'accessibilitat WAVE on es pot comprovar que la pàgina inicial, ja des dels primers moments de desenvolupament, obté 0 errors principals.*
- *Fig. 18: advertiment de "cookies" en compliment del Reglament General de Protecció de Dades (RGPD)*
- *Fig. 19: exemple de control de versions de software. Font: wikipedia*
- *Fig. 20: pantalla inicial de Publii*
- *Fig. 21: aspecte de la finestra de configuració del servidor on publicarem el nostre web*
- *Fig. 22: plantilla SIMPLE per a Publii*
- *Fig. 23: plantilla SQUARE per a Publii*
- *Fig. 24: plantilla TECHNEWS per a Publii*
- *Fig. 25: començament de l'adaptació de la plantilla Square*
- *Fig. 26: exploració del codi de la pàgina utilitzant "Firefox Developer Tools"*
- *Fig. 27: Codi CSS personalitzat*
- *Fig. 28: llistat d'articles*
- *Fig. 29: finestra d'edició d'articles en mode WYSIWYG*
- *Fig. 30: finestra d'edició d'articles en mode codi HTML*

- *Fig. 31: barra lateral d'edició d'articles*
 - *Fig. 32: procés de treball amb Photoshop*
 - *Fig. 33: Inclusió d'una marca d'aigua en un conjunt d'imatges amb PhotoMarks*
 - *Fig. 34: pantalla inicial amb el menú principal desplegat a la dreta*
 - *Fig. 35: pantalla de backups*
 - *Fig. 36: pantalla de sincronització de Publii*
 - *Fig. 37: diferències de representació entre previsualització local i definitiva remota*
 - *Fig. 38: explicacions de Jekyll sobre els requisits per fer-lo funcionar*
 - *Fig. 39: web de baixada de Ruby: <https://rubyinstaller.org/downloads/>*
 - *Fig. 40: instal·lació de Ruby a Windows 10*
 - *Fig. 41: instal·lació de Jekyll*
 - *Fig. 42: estructura inicial del nostre blog fet amb Jekyll*
 - *Fig. 43: pàgina per defecte del nostre nou blog amb Jekyll*
 - *Fig. 44: codi markdown a l'editor markdownguide*
 - *Fig. 45: codi HTML generat a partir del codi markdown*
 - *Fig. 46: visualització final del nostre codi markdown al navegador*
 - *Fig. 47: codi markdown inserta a l'índex del nostre blog amb Jekyll*
 - *Fig. 48: edició d'un arxiu markdown per a Jekyll (bloc de notes de Windows)*
 - *Fig. 49: edició del mateix arxiu anterior fent servir l'editor integrat de Publii*
 - *Fig. 50: nou editor de blocs de la versió 0.36 de Publii*
 - *Fig. 51: nou editor markdown integrat a Publii 0.36*
 - *Fig. 52: resultat de la pàgina "index" amb estructura de 3 contenidors horitzontals iguals*
 - *Fig. 53: pàgina "Estacions FGC" amb llistat de dos contenidors i llistes personalitzades*
 - *Fig. 54: pàgina "Unitats Renfe" amb llistat horitzontal de 4 contenidors de diferents mides*
 - *Fig. 55: la mateixa pàgina de la fig. 51, en versió mòbil*
-
- *Taula 01: planificació*
 - *Taula 02: pressupost*

Canvis des de la versió 3 a la versió final de la memòria

- **Índex:** actualitzat.
- **5. Continguts:** estructura modificada
- **10. Procés de desenvolupament:** es deixa en aquest punt únicament una explicació general de quin és el procés de desenvolupament. Tota la part de configuració, instal·lació, modificació de temes es passen als nous punts 18, 19 i 20
- **17. Versions de l'aplicació:** actualitzat.
- **18. Requisits d'instal·lació:** nou punt on s'expliquen els requisits d'instal·lació
- **19. Instruccions d'instal·lació:** nova explicació ampliada de tot el procés de creació de lloc web i personalització de temes.
- **20. Instruccions d'ús (edició):** nou apartat explicant les instruccions d'ús i edició del sistema de publicació
- **26. Vita Marçal Guardiola Sánchez:** afegit apartat Vita.
- **Annex 2:** extractes de codi font actualitzats.
- **Annex 4:** captures de pantalla actualitzades.
- **General:** reordenació d'apartats i numeració per a la versió definitiva de la memòria. Correccions menors en la documentació de diversos apartats.

1. Introducció

En aquest treball que ens ocupa volem partir d'una pàgina web existent i fer-ne una remodelació per modernitzar-la i adaptar-la als estàndards actuals.

Es tracta d'una pàgina web personal (<http://www.wefer.com>) dedicada a donar a conèixer la història i evolució del ferrocarril a Espanya i que en el seu moment, a finals dels anys 90, va ser una pàgina web de referència, però que actualment ha quedat superada per altres pàgines web més recents i pel contingut existent a les xarxes socials.

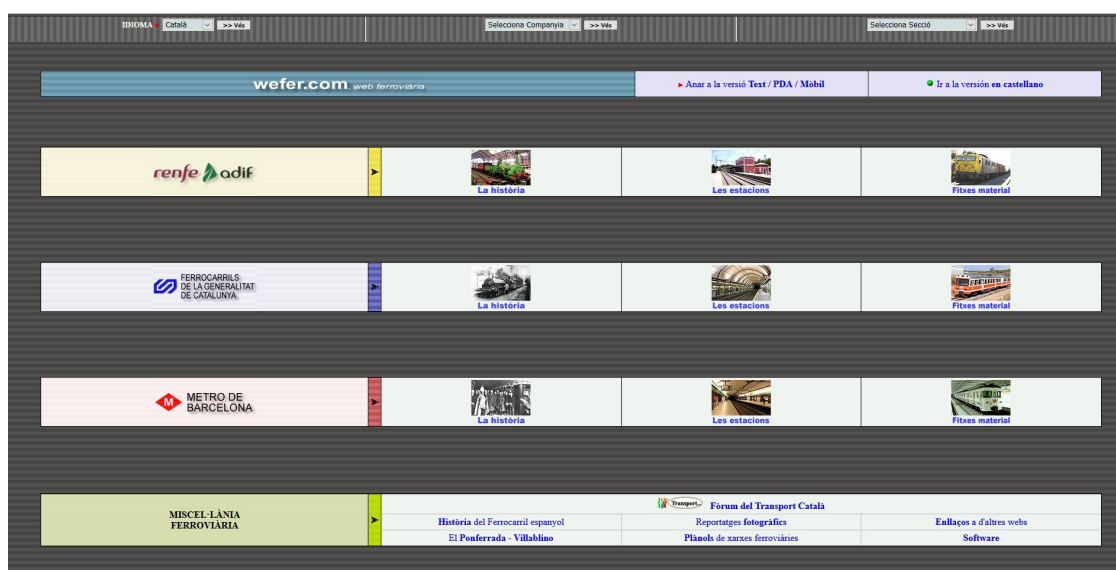


Fig. 01: aspecte actual de la pàgina inicial, realitzada amb taules i "frames"

Donada la seva antiguitat, aquesta pàgina web va ser dissenyada amb la tecnologia existent en aquell moment que consistia en dissenyar les pàgines codificant-les directament utilitzant **HTML sense la utilització d'estils en cascada**. És a dir, la definició formal d'estils es trobava inclosa i barrejada amb l'estructura HTML. A més, s'utilitzaven elements que en aquell moment eren innovadors però que avui dia no només han quedat superats per la tecnologia actual, sinó que a més estan totalment desaconsellats i possiblement seran eliminats de l'estàndard HTML en el futur, com per exemple la utilització de "**frames**" per dividir la pantalla, obsolets en HTML5.

La primera idea de remodelació ens porta, per inèrcia, a la utilització dels sistemes més habituals avui dia per a la realització de pàgines/blogs, amb **WordPress** com a element estrella i **Joomla!**, **Drupal** o similars com alternatives.

No obstant, si ens parem a pensar en les necessitats reals de la pàgina, veiem que al tractar-se d'una **pàgina informativa documental**, malgrat l'antiguitat de la seva tecnologia la pàgina funciona de manera suficientment satisfactòria per la qual cosa fa pensar que no cal una remodelació total sinó només adaptar aquells elements més

arcaics a les noves necessitats: depreciació d'elements obsolets, **adaptació responsive**, etc. Dit d'una altra manera, gran part de les possibilitats dinàmiques que ofereixen els CMS actuals no són realment necessàries.

Aquest fet fa **reflexionar si hem de seguir la via convencional** d'instal·lar un **CMS actual** o bé existeixen mètodes alternatius que poden funcionar igual de bé i que, fins i tot ens poden donar algun avantatge no pensat en un primer moment.

Aquest fet m'ha portat a investigar quines són les alternatives que avui dia existeixen per poder desenvolupar un web documental descobrint les següents opcions principals:

- **Desenvolupament des de zero:** es tracta de la creació d'un lloc web dissenyant-ne les pàgines una a una a l'antiga, bé sigui directament creant el codi a bé ajudant-nos de programes WYSIWYG com **Dreamweaver** que ens permeten controlar una gran quantitat d'arxius estructurats i actualitzar-los a la xarxa de manera manual. Es tracta del mètode clàssic de disseny però que cada vegada s'utilitza menys, com a mínim en el camp dels llocs webs petits, ja que implica molt treball de codi per a resultats que es poden aconseguir més fàcilment amb els altres sistemes comentats.
- **Flat File CMS:** es tracta de CMS amb les mateixes característiques que els CMS habituals (Wordpress, Drupal, Joomla!, etc) però que **no fan servir base de dades** sinó que desen tota la informació en arxius de text plans. No obstant, les pàgines són generades dinàmicament mitjançant PHO (o altres) igual que en els CMS tradicionals. En principi tenen l'avantatge de ser més senzills de gestionar i fer backups ja que tot es desa dins de l'estructura de directoris del web i no en un servidor de BBDD a part. De tota manera la rapidesa es pot arribar a veure compromesa pel fet d'haver de generar les pàgines com qualsevol altra CMS convencional. Un exemple d'aquest sistema és **GRAV**.
- **Static Sites CMS:** es tracta de CMS en els quals les pàgines ja estan generades en HTML (i habitualment Javascript) de manera que el servidor només les ha de servir al client. **Les pàgines són generades en local** per part del creador del web i una vegada acabades es penjen en el servidor ja completades. En aquest cas la **rapidesa és extrema** comparada amb un CMS convencional (aproximadament **600% més ràpid que un Wordpress convencional**. També té avantatges en quant a la seguretat ja que no fa servir mètodes que puguin utilitzar els hackers per entrar al sistema, com ara injeccions SQL.

Per aquest treball i tenint en compte tot l'explicat, s'ha optat per fer servir el **programari Publii** per crear un lloc web estàtic i gestionar-ne el manteniment.

2. Descripció

Les raons d'haver escollit realitzar un lloc web estàtic són les següents:

- Major **rapidesa** degut a que el codi no ha de ser executat sinó únicament interpretat.
- **Senzillesa** de disseny a través d'aplicacions en local.
- Senzillesa de realització de **backups** i portabilitats entre servidors ja que només cal que copiem els directoris amb un FTP clàssic i ja tenim el web sencer incloent les dades.
- Major **seguretat** ja que no depenem de les constants actualitzacions de seguretat de les tecnologies actives (PHP, MySQL, etc) i evitem forats d'entrada com les injeccions SQL.

Totes aquestes premisses s'adapten perfectament al que volem aconseguir amb la remodelació d'aquesta web.

wefer.com és una pàgina web dedicada a l'estudi i divulgació del món ferroviari a Catalunya. Com a tal, té un esperit documental que encaixa perfectament amb l'essència d'un lloc web estàtic. L'estructura resumida del lloc web serà:

- **Renfe/ADIF**
 - **La història:** recull històric de les fites més importants. El contingut és principalment text enriquit amb unes poques imatges i taules. Es troba dividit en diverses pàgines segons èpoques
 - **Les estacions:** és l'apartat més extens. Amb el temps, inclourà un llistat de totes les estacions de la xarxa a Catalunya. Cada estació té una pàgina web on es mostren fotografies, textos, etc. En el nou web també seran accessibles a través d'un mapa geolocalitzat.
 - **El material:** llistat de material. Cada tipus de material té la seva pàgina única amb fotografies i dades. En la nova versió, s'hi podran incloure vídeos i altra material disponible a les xarxes socials.
- **Ferrocarrils de la Generalitat de Catalunya** (*aparts homòlegs a Renfe/ADIF*)
 - **La història**
 - **Les estacions**
 - **El material**
- **Metro de Barcelona** (*aparts homòlegs a Renfe/ADIF*)
 - **La història**
 - **Les estacions**
 - **El material**

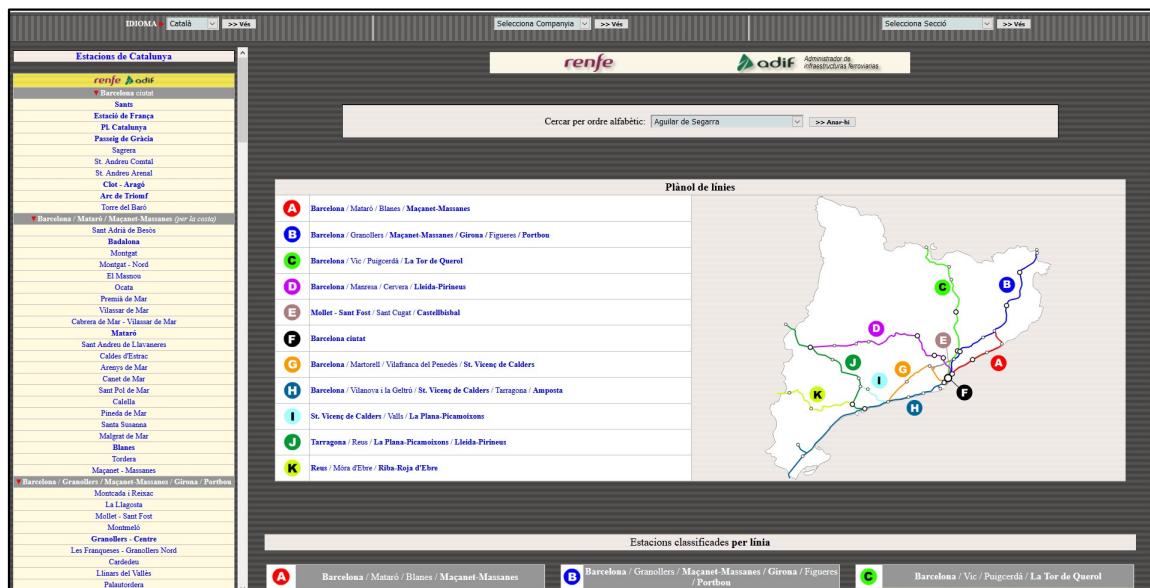


Fig. 02: aspecte actual de la pàgina d'estacions a remodelar

- **Història del ferrocarril a Espanya:** l'estructura serà similar als apartats "La història" abans comentats, però aquí es farà una visió general de la història del ferrocarril comprenent èpoques anteriors.
- **Reportatges fotogràfics:** aquí s'inclouran col·leccions de fotografies de temes ferroviaris que no es puguin incloure en altres apartats.
- **Plànols:** recull d'imatge de plànols i les seves explicacions.
- **Altres:** tot allò que no tingui cabuda en els anteriors apartats.

3. Objectius

3.1. Objectius principals

- **Investigar** les possibilitats, **avantatges i inconvenients** dels "static sites" dins del web actual.
- **Investigar** les possibilitats de **disseny** de "static sites" utilitzant un sistema "WYSIWYG" com **Publii**.
- **Quantificar aquests avantatges** en forma de millores en la velocitat i/o seguretat del sistema final.
- En cas que les proves siguin satisfactòries, **publicar el resultat** obtingut a la xarxa i substituir el web existent actualment.
- En cas que les proves fracassin, **detectar quins han estat els problemes** i proposar solucions (en cas d'existir) als creadors del sistema.

3.2. Objectius secundaris:

- **Modernitzar una pàgina web** que havia estat la referència del seu sector.
- Crear una web que **compleixi els estàndards** tècnics actuals com el disseny responsive.
- Muntar un **sistema fàcil i ràpid d'actualització** del contingut.
- Integrar serveis web dinàmics externs dins de l'estructura web estàtica, tal com les **xarxes socials**.

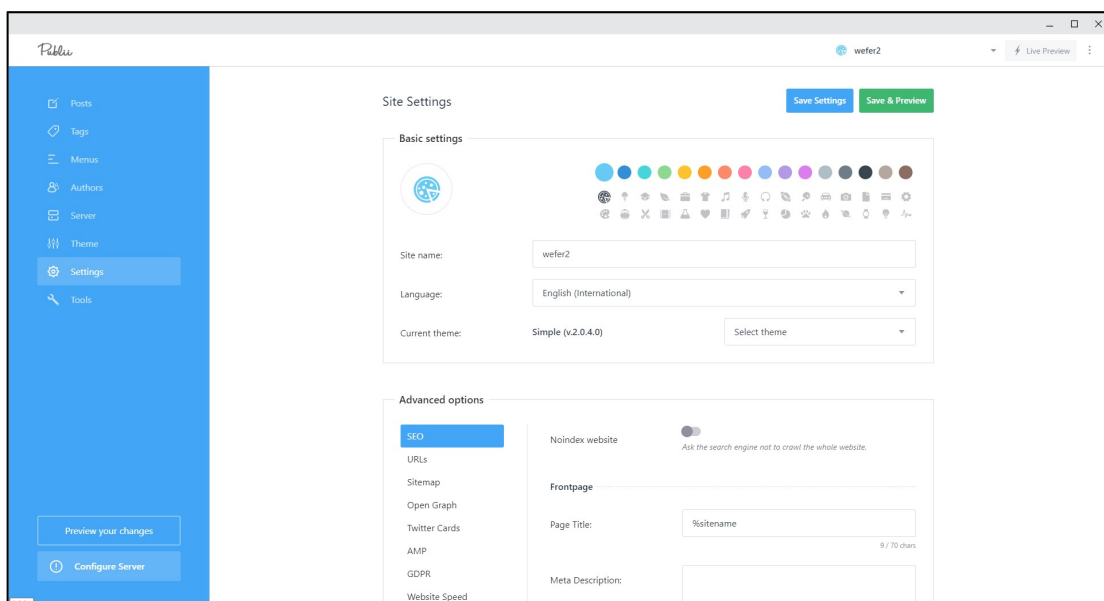


Fig. 03: interfície de l'aplicació Publii a Windows 10

4. Marc teòric

En els primers anys de popularització d'internet entre el públic en general, a finals dels anys 90 del segle passat, es van començar a crear pàgines web senzilles l'únic objectiu de les quals era mostrar un text i algunes imatges (poques) als lectors.

La majoria eren **pàgines informatives estàtiques**, a les quals s'accedia a través de directoris realitzats manualment. La tecnologia existent no permetia gaire cosa més i tampoc els amples de banda disponibles

Aviat es va veure que per a moltes aplicacions era necessari crear **pàgines web dinàmiques** que interactuessin amb l'usuari i responguessin a les seves peticions, sobretot en el camp de l'incipient comerç electrònic, la tecnologia de cercadors i, més endavant, les primers aplicacions multimèdia.

Paral·lelament, la tecnologia i potència dels servidors i l'ample de banda disponible feien possibles noves maneres interactives de crear la web.

Poc a poc la WWW es va omplir de llocs dinàmics, es va anar deixant de banda l'**HTML** clàssic deixant pas a llenguatges com **Javascript**, **PHP**, etc que si ve es continuaven basant en HTML permetien una web molt més interactiva.

Per explotar al màxim les possibilitats d'interacció, aquests llenguatges es van integrar amb les noves cascades d'estils **CSS** i l'accés a bases de dades com **MySQL**, entre d'altres.

Així, vam passar d'una web estàtica i senzilla que requeria poc poder de processament a una plena d'interacció que avui dia demanda **ingents quantitats de potència** i ample de banda.

No obstant, arribats a aquest punt, cal que parem un moment per pensar si realment totes les aplicacions actuals necessiten realment tota la tecnologia que fan servir, o bé si en alguns casos estem caient en la **sobreenginyeria** i en el malgastament de recursos.

És evident que un cercador modern com Google, un lloc multimèdia com Youtube o una xarxa social com Facebook necessiten grans dosis d'interacció i tecnologia que no poden ser oferts per HTML, tot i l'evolució que ha fet en els últims anys.

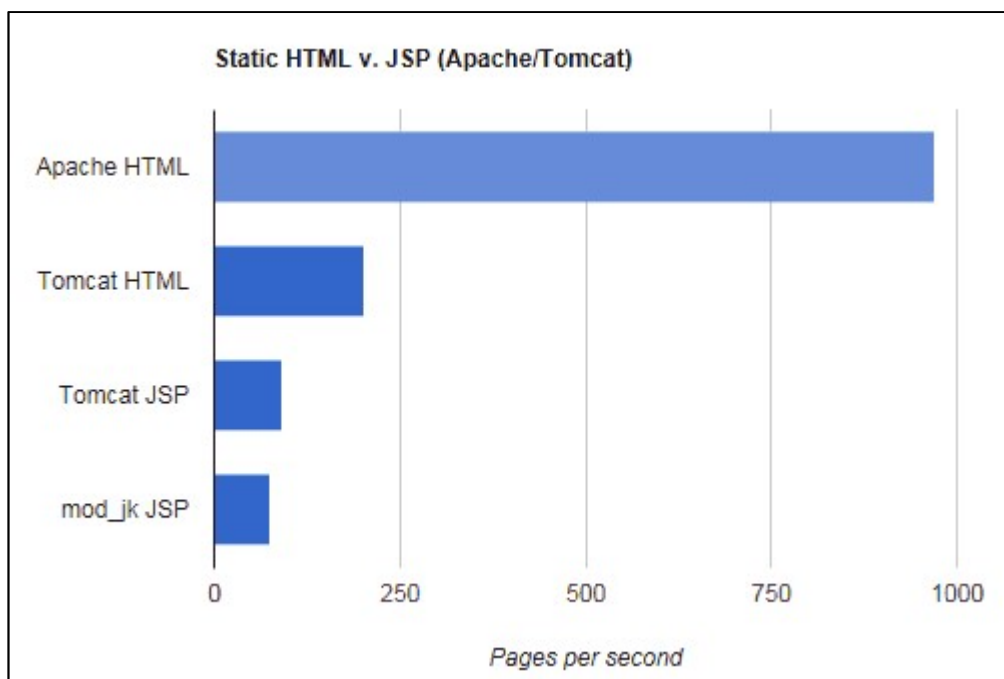


Fig. 04: comparació de velocitat entre HTML i el llenguatge dinàmic JSP (Java Server Pages). Font: "A Web Coding blog"

Però realment, **un blog personal** on es mostraran alguns textos i fotografies i s'integrarà algun vídeo publicat a Youtube **¿necessita una gran infraestructura de llenguatge de PHP, bases de dades MySQL, etc, amb el que això implica de consum de recursos, dificultat de manteniment i potencials forats de seguretat?**

El treball que ens ocupa vol cercar fins a quin punt és factible **tornar parcialment a una web més senzilla**, que demandi menys recursos, que sigui més fàcil de mantenir i en conseqüència molt més barata, però que no signifiqui renunciar a una web **moderna** que satisfaci les necessitats de la majoria de petits usuaris.

5. Continguts

La nova pàgina wefer.com segueix l'esquema de l'anterior, que es caracteritza per una **estructura poc jerarquizada** i força horitzontal, de manera que des de la pàgina inicial es pot accedir a tots els apartats principals i aquests, a la vegada, permeten accedir al contingut final en la majoria dels casos.

D'aquesta manera, en **tan sols dos clics** som capaços d'arribar al contingut principal de la pàgina, i amb un tercer clic arribem a la part més profunda de l'estructura com pot ser cadascuna de les fotografies d'un carrusel fotogràfic.

En la pàgina inicial ens trobem amb una de les primeres limitacions de Publii: és un sistema que **no permet la construcció de llocs web multiidioma**, per la qual cosa en aquest cas s'ha optat per començar fent una sola versió (la més visitada, en castellà) a l'espera que **aquest mateix any 2020** (segons els seus autors) s'incorpori aquesta nova opció i poder **afegir una versió en català** i una altra en anglès del mateix lloc. Com que en aquests moments no sabem com es farà la implantació del mode multiidioma, és impossible saber si caldrà canviar l'estructura en la seva part superior per incloure un selector d'idioma o bé es farà **de forma automàtica** a partir de la configuració del navegador de l'usuari. Aquesta darrera opció seria la preferida ja que seria transparent per a l'usuari i no implicaria canviar l'estructura.

Com podem veure a l'esquema de la pàgina següent, el lloc web s'estructura en **quatre grans bloc principals**, els tres primers formats per cadascuna de les companyies ferroviàries a les que es refereix el lloc web i un tercer bloc de miscel·lània on s'encabeixen la resta d'apartats. Veure **punt 2 (Descripció)** per una explicació més concreta dels continguts.

A totes les pàgines s'inclou en primer lloc un "**breadcrumb**" de manera que l'usuari pugui saber en quin punt de l'estructura es troba, així com poder tornar fàcilment a qualsevol dels graons superiors de la cadena.

De la mateixa manera, a l'esquerra de la pàgina s'inclou en totes les pàgines **el logotip de la web** que permet accedir directament a la pàgina inicial.

Dins de cada pàgina, hi ha tant text com imatges incrustades. Una part molt important de la web són les àmplies **col·leccions de fotografies** de cadascun dels elements que es tracten (bàsicament estacions i material ferroviari). Aquest material gràfic és visible en els **carrousels fotogràfics** abans esmentats, sobreposats com una capa sobre la pàgina principal d'informació de l'element.

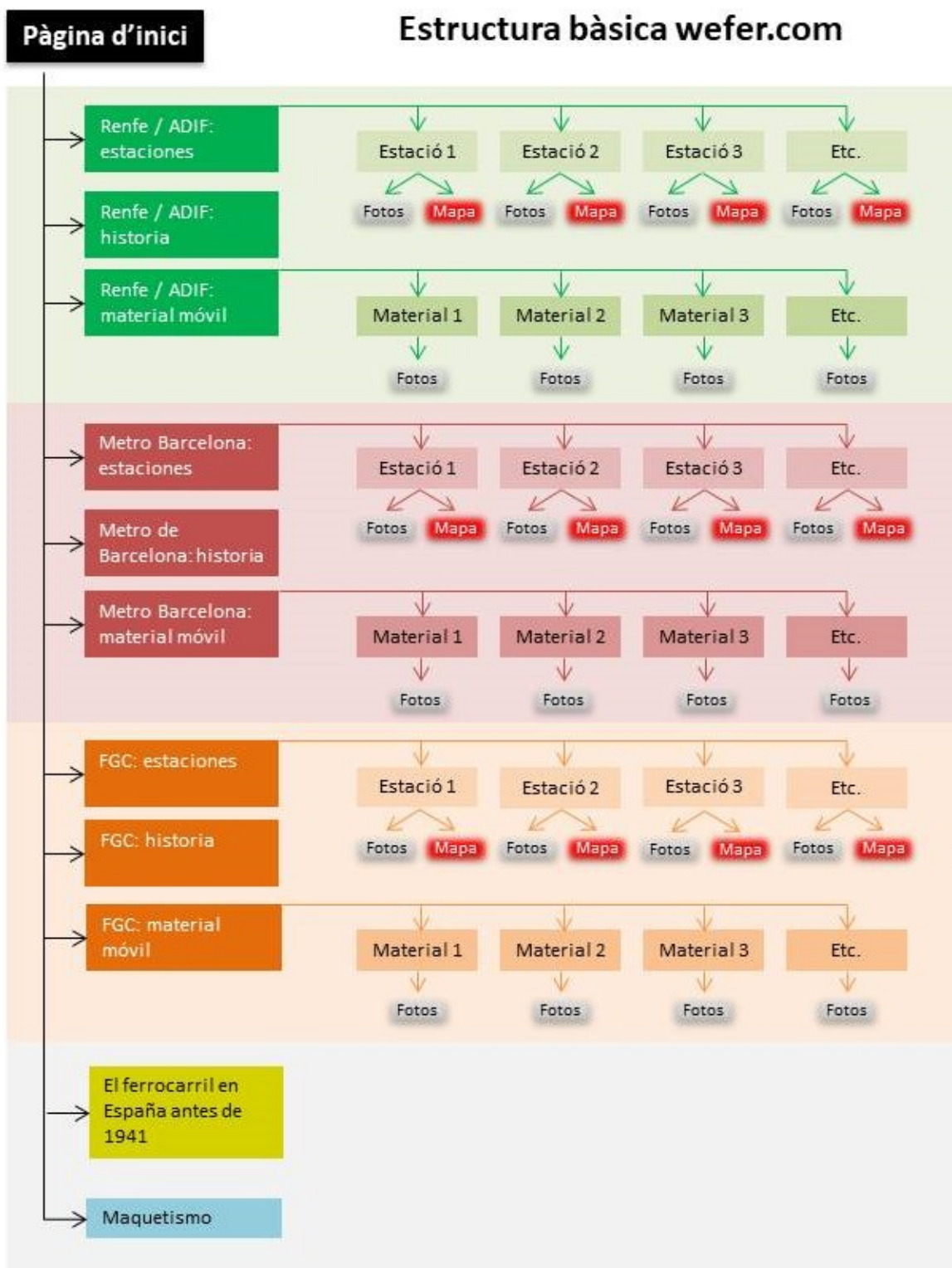


Fig. 05: estructura de wefer.com

6. Metodologia

Per tirar endavant aquest projecte s'han tingut en compte dos enfocaments possibles: utilitzar un esquema "*agile*" o bé fer servir el clàssic en cascada.

En el cas que ens ocupa, tenim uns requisits força definits i a més es tracta d'un projecte relativament petit. A més, el fet que tot el web hagi de ser dissenyat per una sola persona que, a la vegada, n'és el client, fa que la senzillesa en la gestió sigui primordial.

Per tot plegat, s'ha optat per un sistema clàssic de **treball en cascada** en el qual les tasques es realitzen seqüencialment una rere l'altra.

Les principals tasques que es realitzaran seran:

- Creació dels **wireframes** del nou web.
- Definició de l'**estructura**. S'ha creat l'estructura de menús del nou web en base a l'existent.
- Creació d'un **web provisional** on s'han anat penjant els canvis. Donat que una de les funcions principals que s'han provat en aquest projecte són la sincronització entre els arxius locals i un servidor extern, a més de la migració entre servidors, s'ha decidit anar publicant en un servidor provisional i amb un domini provisional tots i cadascun dels canvis, en comptes de publicar localment fent servir un entorn XAMPP o similar.
- Configuració i posada en marxa de l'aplicació **Publii** en l'ordinador de disseny.
- Realització dels elements que configuren la **nova imatge** del lloc web.
- Creació, una a una, de les **pàgines** que conformen el nou web.
- Col·locació dels **textos i imatges** corresponents provinents de l'antic web.
- **Adequació d'estil** i formal de tot el contingut.
- Inclusió de les **noves prestacions**, com la integració amb xarxes socials.
- **Comprovació** i depuració del conjunt.
- Trasllat al **servidor final**.
- Realització de l'informe de **conclusions** sobre el funcionament del sistema Publii i del resultat obtingut.

7. Arquitectura de l'aplicació

L'arquitectura de l'aplicació que ens ocupa és la típica d'un servidor web senzill capaç de servir fitxers HTML:

- **Client:** cal que disposi d'un navegador d'internet relativament modern, capaç d'interpretar tant HTML (amb CSS) com JavaScript. Qualsevol navegador modern compleix perfectament aquests requisits.
- **Servidor:** el servidor ha de ser capaç de servir pàgines HTML. No ens cal cap altre llenguatge. A més, per a l'actualització, cal que ens puguem connectar a través de FTP, preferentment de manera segura amb SFTP. Ambdós requisits són satisfets per qualsevol servidor existent avui dia.

No necessitem bases de dades ni llenguatges dinàmics (com PHP) en cap moment.

El funcionament bàsic de la pàgina web és l'explicat en la següent figura:

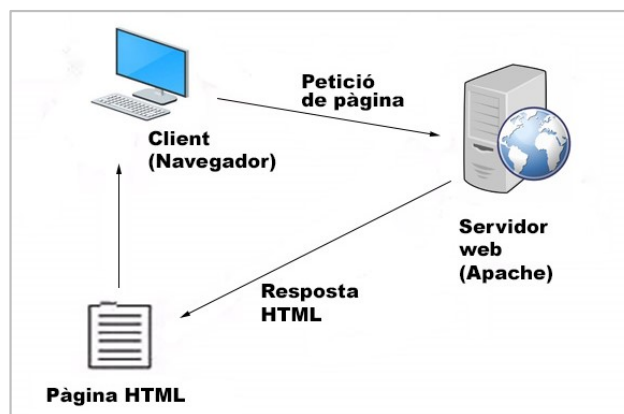


Fig. 06: funcionament esquemàtic d'una petició de pàgina web per part d'un client

Per poder tenir el nostre espai web tan sols haurem de **donar-nos d'alta en un proveïdor de "hosting"** a internet. Aquest ens proporciona una interfície web (per exemple PLESK) des de la qual gestionar el nostre espai.

No obstant, en el cas que ens ocupa la pàgina ja existeix i està allotjada en un servidor virtual compartit amb altres llocs web.

Una vegada acabem la fase de desenvolupament, únicament ens cal esborrar tot el contingut de la web antiga, ja sigui utilitzant un programa FTP com FileZilla o els propis recursos de PLESK per deixar totalment net el director principal (httpdocs).

És important que quan pugem el lloc en el servidor de producció fem aquest pas d'esborrar els arxius antics ja **que Publii sobreescriu els fitxers que tinguin el mateix**

nom i puja els que no existeixen, però en cap cas elimina els ja existents que no es trobin a l'estructura del lloc Publii. Si no eliminem aquests fitxers quedaran sense ús ocupant espai i, probablement, seguiran generant tràfic a través dels buscadors que dirigiran a pàgines trencades.

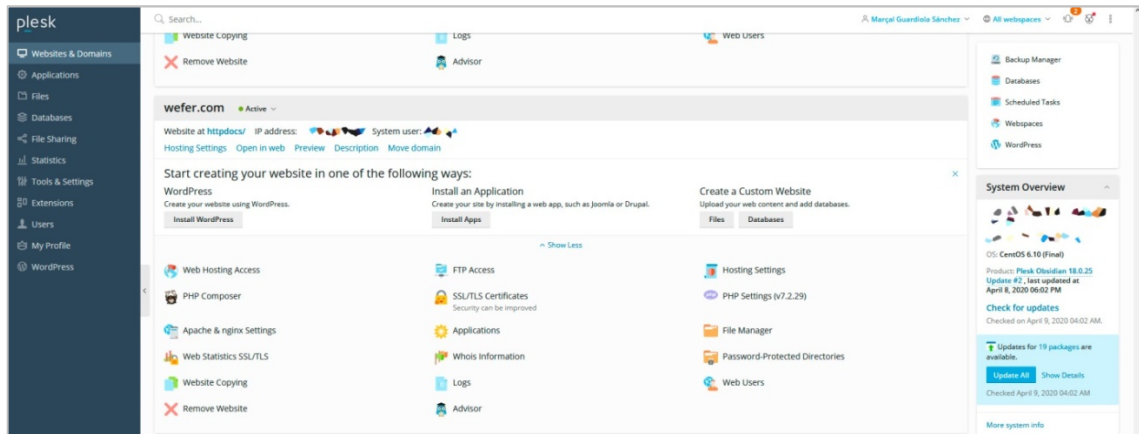


Fig. 07: PLESK, un dels gestors que ens permet controlar els paràmetres del nostre lloc web

En aquest cas tampoc ha calgut prendre cap acció pel que fa al nom de domini, ja que ja es troba enllaçat al servidor i a l'espai web corresponent.

El nom wefer.com es troba registrat en un registrador de domini extern al del hosting, per la qual cosa en la configuració d'aquest registrador estan donades d'alta les adreces DNS del proveïdor de hosting que ens proporciona el nostre servidor web que conté la pàgina per tal que la redirecció del nom cap a la IP del nostre servidor es realitzi correctament.

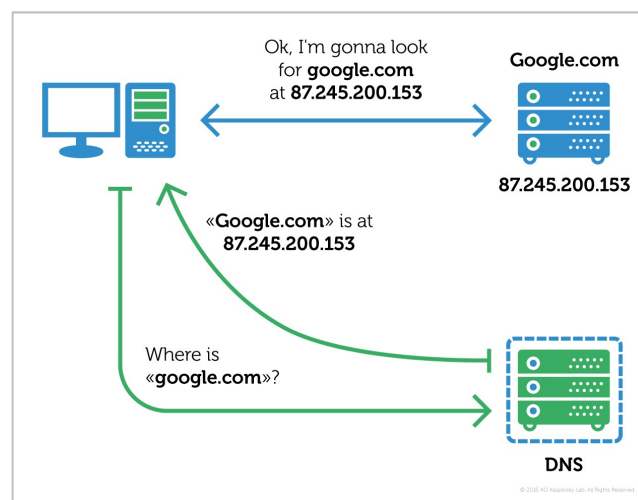


Fig. 08: funcionament d'un servidor DNS per a la resolució de noms de domini a internet. Font: kaspersky.es

8. Plataforma de desenvolupament

- **Maquinari local:**
 - **PC** amb processador Intel i5 a 2.67 GHz, 10 GB de RAM, disc principal SSD de 500 GB i secundari mecànic de 1 TB. Targeta gràfica Nvidia GTX1050.
 - **1 monitor LED de 23 polzades (Wide)** a 1920 x 1980 píxels.
 - Connexió a internet a través de cable Ethernet (Gigabit) fins a router per a **fibra de 600 MB** simètrics.
- **Programari local:**
 - Sistema operatiu **Windows 10** com a mètode principal de desenvolupament
 - Sistema operatiu **Linux Mint** com a plataforma per provar la versió Linux del programa Publii.
 - **Publii**: base del treball actual, ens permet generar el lloc web estàtic en local i quan el tenim acabat pujar-lo al web.
 - **Adobe Photoshop CS6**: per al tractament d'imatges, generació de GIFs animats i esquemes.
 - **Adobe Dreamweaver CS6**: en alguns casos com l'edició CSS o la creació de DIVs les eines WYSIWYG de Publii són insuficients i/o generen resultats no desitjats. en aquests casos la solució ha estat ajudar-se de Dreamweaver per tal de crear el codi i posteriorment enganxar-lo a Publii en format HTML.
 - **Photomarks**: programa d'edició d'imatges en lot. Permet realitzar tasques complexes com l'adició de marques d'aigua, amb controls de posició avançats, de manera ràpida a centenars o milers d'imatges a la vegada.
 - **FileZilla**: client FTP utilitzat puntualment per tal de fer neteja d'arxius que ja no s'utilitzen.

S'ha fet servir un **servidor de prova** per allotjar un espai web amb un domini servidor de prova (magsx.com) idèntic al de producció final i allotjat en el mateix servidor. D'aquesta manera podem fer les pujades al web de prova que vulguem mantenint el web antic en línia fins el moment en que tinguem 100% acabat el nou web, fent que estigui disponible en la seva totalitat en tot moment.

- **Maquinari del servidor de prova:**
 - **Servidor virtual** amb processador AMD Opteron, 8,8 GB de RAM i 281 GB d'espai en disc.
- **Programari del servidor de prova:**
 - Sistema operatiu **Linux CentOS 6.10**.
 - Gestió d'espais web **Plesk Obsidian**.
 - Gestió de servidor **Parallels Power Panel**.
 - Servidor web HTTP **Apache 2.2.15**.

9. Planificació

Id tasca	Nom de la tasca	Inici	Final
1	Definició del projecte	18/02/2020	29/02/2020
2	Document preliminar d'objectius i requisits	01/03/2020	03/03/2020
3	Entrega PAC1		03/03/2020
4	Realització dels wireframes	04/03/2020	07/03/2020
5	Definició formal	08/03/2020	10/03/2020
6	Definició de l'estructura	11/03/2020	14/03/2020
7	Posada en funcionament del software de desenvolupament	15/03/2020	16/03/2020
8	Muntatge del servidor de prova	17/03/2020	18/03/2020
9	Creació de les noves imatges	19/03/2020	31/03/2020
10	Creació de les pàgines i de l'estructura	21/03/2020	31/03/2020
11	Còpia dels primers textos	21/03/2020	31/03/2020
12	Preparació segon informe d'entrega parcial	01/04/2020	02/04/2020
13	Entrega PAC2		03/04/2020
14	Finalització de la còpia de continguts originals	04/04/2020	16/04/2020
15	Inclusió dels mitjans gràfics	17/04/2020	22/04/2020
16	Inclusió dels mitjans multimèdia	17/04/2020	22/04/2020
17	Ajustament estilístic de tot el conjunt	23/04/2020	29/04/2020
18	Preparació del tercer informe d'entrega parcial	30/04/2020	02/05/2020
19	Entrega PAC3		03/05/2020
20	Comprovació de tot el conjunt en el servidor de prova	04/05/2020	14/05/2020
21	Versió Beta		15/05/2020
22	Correccions d'errors	16/05/2020	26/05/2020
23	Còpia al servidor definitiu	27/05/2020	28/05/2020
24	Comprovació del resultat final	29/05/2020	02/06/2020
25	Confecció del document final de conclusions	03/06/2020	14/06/2020
26	Lliurament final		15/06/2020

Taula 01: planificació

Diagrama de Gantt:

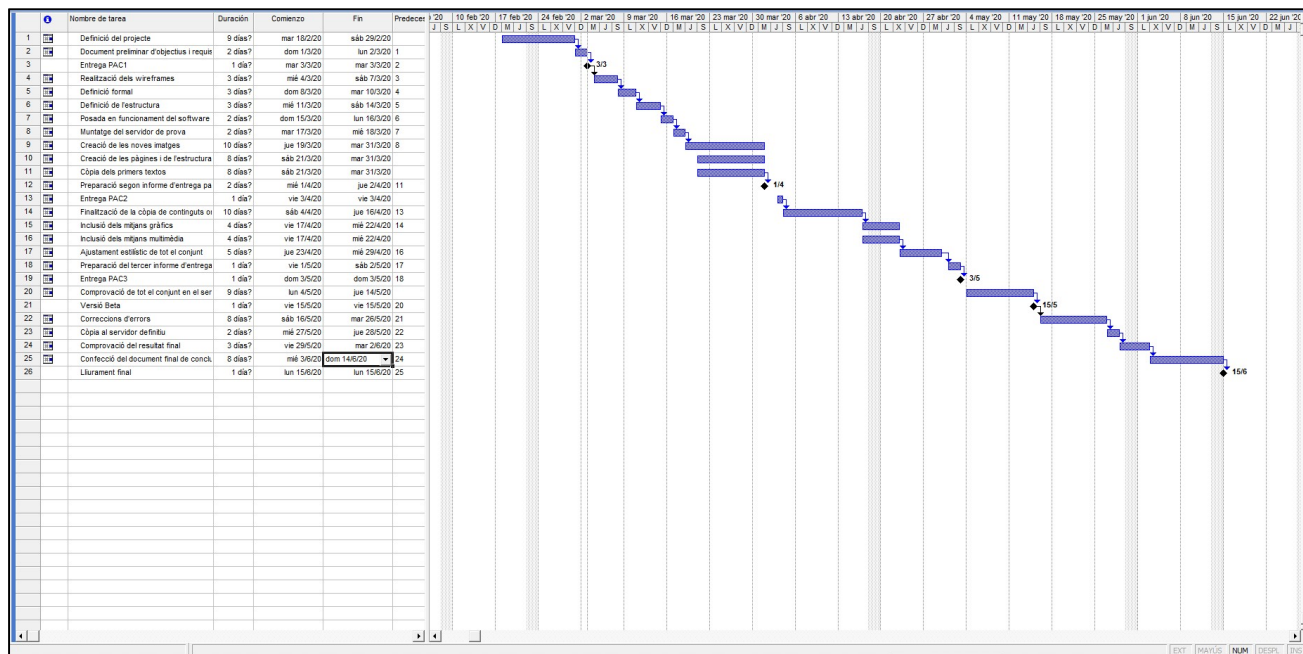


Fig. 09: temporització del projecte (diagrama de Gantt)

10. Procés de desenvolupament

El procés de treball amb Publii per generar un **lloc web estàtic** és diferent al que feríem servir si féssim servir un CMS convencional basat en bases de dades, com poden ser Wordpress, Joomla! o Drupal entre d'altres.

En els **CMS convencionals** tenim un **backoffice** en el qual fem els canvis al nostre lloc web, i un **frontoffice** que és el que veu l'usuari que accedeix a la nostra publicació. Ambdós es troben en el mateix servidor però en URLs diferents, de tal manera que els canvis del backoffice són immediats en el frontoffice. Això, per a temes de desenvolupament, ens obliga a treballar sobre un servidor de prova per fer canvis profunds (més enllà d'editar un simple post) i copiar els canvis quan aquests estiguin acabats i funcionin correctament. Com s'ha explicat anteriorment això genera una gran complexitat de servidors, bases de dades, còpies de seguretat, etc.

El procés de treball amb **Publii** és diferent, ja que d'alguna manera podem considerar que el **frontoffice es troba en el servidor** de producció, mentre que el **backoffice es troba en un ordinador local** que no cal que tingui res més instal·lat que el propi Publii i un navegador web. El procés resumit és el següent:

- **Obrim Publii** en el nostre ordinador local.
- **Escollim el lloc web** que volem editar.
- **Personalitzem el lloc web** (configuració, plantilles, codi personalitzat, etc)
- **Creem les pàgines** i les omplim de contingut (textos, imatges, tags, enllaços, etc).
- Amb el mateix Publii **renderitzem les pàgines** de prova per veure com queden. Poden utilitzar el botó "Preview" de cada post per veure com quedarà el post que estem editant o bé utilitzar el botó "Preview your changes" per renderitzar localment tot el lloc web i fer les proves pertinents.
- Una vegada tenim tots els elements correctes i funcionant, cliquem el botó "**Sync your website**" i els canvis es puguen al servidor automàticament al servidor

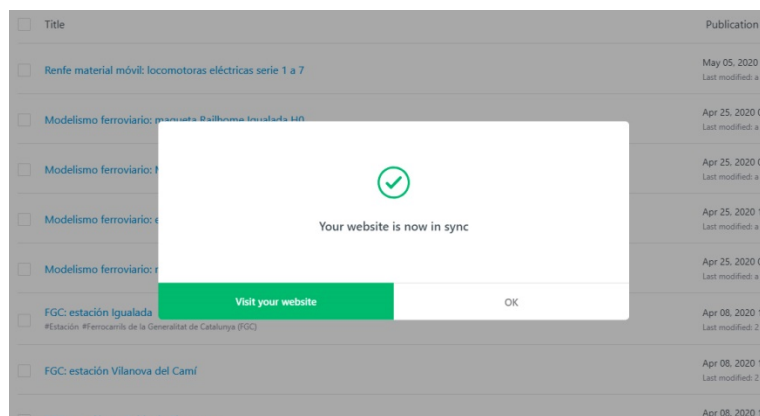


Fig. 10: finalització el procés de sincronització

Aquesta manera de fer el desenvolupament amb Publii presenta dos inconvenients respecte el treball amb els CMS convencionals:

- Cada vegada que es fa un canvi **cal sincronitzar tot el lloc web**. Sincronitzar no significa que pugi tot el contingut cada vegada sinó que cal que Publii detecti quins fitxers s'han canviat i els substitueixi al servidor. En llocs web grans i/o amb connexions lentes pot suposar un temps considerable.
- La renderització en local **no sempre és exacte**. Hi ha elements que es mostren diferents una vegada són pujats al servidor (veure apartat 23. Bugs). Possiblement es deu al fet que l'aplicació encara es troba en fase beta.

Per altra banda, també presenta una sèrie d'avantatges:

- Ens permet fer tots els canvis que vulguem, de la profunditat que vulguem, i provar-los dins el mateix programa.
- **Publii integra un gestor de backups** de manera que podem trastejar el nostre lloc web tant com vulguem en local ja que, en cas que no ens convencin els canvis, amb uns pocs clics podem restaurar l'estat anterior. De fet, simplement copiant la carpeta on es desen els arxius del web ja tenim una còpia sencera de tot el lloc web llesta per treballar amb Publii.
- De la mateixa manera podem fer diverses còpies del nostre lloc web simplement copiant la carpeta d'arxius i utilitzar-los, per exemple, per fer proves o per tenir diverses versions.
- **Rapidesa d'actualització** de cara a l'usuari ja que l'actualització és immediata i no requereix deixar el lloc web fora de línia en cap moment per molt complexa que sigui l'actualització.
- El resultat final és **un lloc web espectacularment ràpid** degut a que la renderització la fa el desenvolupador una sola vegada (d'aquí la lentitud de sincronització) i no el servidor cada vegada que s'ha de servir la pàgina, estalviant temps i recursos.

En el cas concret que ens ocupa al tractar-se d'una remodelació total d'un lloc web preexistent no realitzat amb Publii, i a més tenint en compte que es tracta d'un treball de comprovació del funcionament del propi Publii, sí s'ha considerat necessari pujar els canvis primer a un servidor diferent allotjat a l'adreça <https://www.magsx.com>. Aquest servidor de prova, no obstant, ja no és necessari a partir del moment en que s'acabi el desenvolupament inicial i es publiqui la primera estructura generada per Publii en el servidor de producció sota l'adreça <https://www.wefer.com>

11. APIs utilitzades

Per realitzar el nou lloc web wefer.com s'han utilitzat dues interfícies de programació:

- **Adobe Dreamweaver:** en aquest programa s'ha creat una part del codi, bàsicament el que es representa a les pàgines que contenen l'accés als articles finals. Aquestes pàgines necessitaven d'una programació en HTML i CSS que anava més enllà dels editors que posseeix Publii ja que, si bé aquests editors integrats permeten fer el que es vulgui, no proporcionen cap mena d'ajuda al programador per la qual cosa són farragosos i difícils de gestionar.

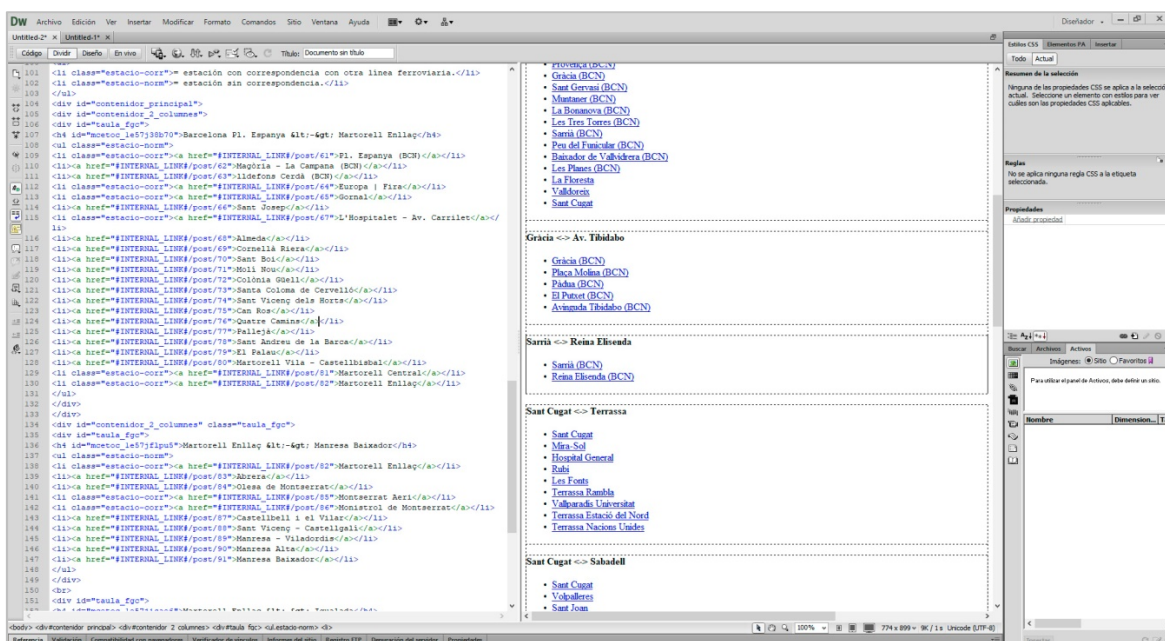


Fig. 11: edició HTML amb dremweaver

- **Editors interns de Publii:** aquest programa té bàsicament dos editors:
 - **Editor HTML:** consisteix en un simple quadre de text sense cap botó de comandament (només salvar). S'accedeix individualment des de cada article i serveix per revisar i modificar exclusivament el codi HTML de l'article seleccionat. No obstant, des de l'apartat "Tools" tenim un editor d'iguals característiques que ens permet col·locar un codi HTML personalitzat en tots els arxius, ja sigui al HEAD, al BODY o al FOOTER. En aquest projecte s'ha utilitzat per enganxar codi provinent de Dreamweaver o per retocar petites parts que no eren possibles de fer amb l'editor visual de text, sobretot pel que fa a la creació de DIVs.
 - **Editor CSS:** igual que l'editor HTML no disposa de cap botó que ens ajudi a insertar codi i tampoc presenta com sistema d'autocompletat que ens ajudi en la nostra tasca de crear noves característiques CSS. Tot el que s'introdueix en aquest editor sobreescrivia o complementa les regles descrites en els CSS principals de la plantilla utilitzada. S'ha utilitzat igual que en el cas anterior per

copiar regles CSS de Dreamweaver, modificant estils d'objectes prèviament analitzats amb les les "Firefox Developer Tools" de Firefox (veure apartat 10.4)

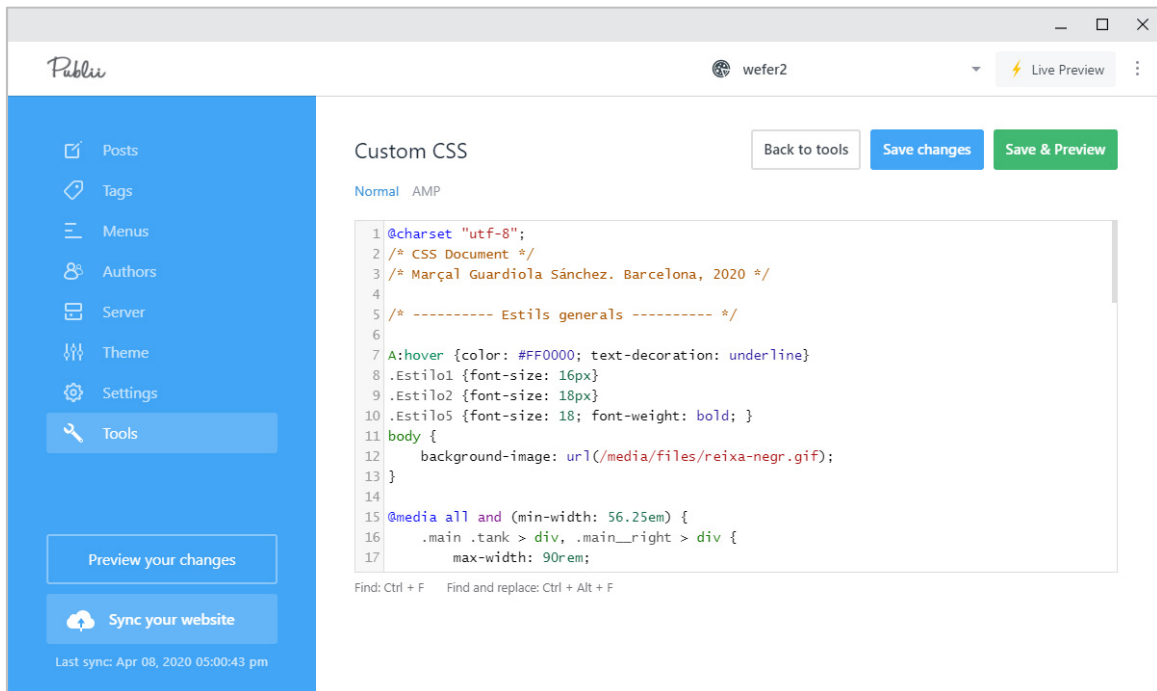


Fig. 12: editor CSS intern de Publii

12. Prototips

Per realitzar el nou lloc web es partia de dos conceptes inicials que calia combinar.

Concepte 1: es volia mantenir al màxim l'estructura original de la pàgina però modernitzant-la tant tècnicament com estèticament. El fet de mantenir l'estructura obligava a mantenir la distribució original de les imatges que donen pas a les seccions principals, totes elles accessibles des de la pàgina inicial.

Per això es va realitzar un primer prototip de prova del que podria ser una versió que remodelava la part tècnica però mantenien una estètica continuïsta. Eliminava els **antiquats "frames"** de HTML i passava a una estructura HTML+CSS responsive basada en DIVs:



Fig. 13: prototip fet en HTML+CSS

Una vegada realitzat aquest prototip es va veure que l'estètica era excessivament continuïsta. A més, una vegada repassat el catàleg **de plantilles disponibles** actualment a Publii (veure apartat 10.4) cap d'ells s'ajustava suficientment a l'estructura demanada.

Donat que ni per temps ni per la corba d'aprenentatge el desenvolupament actual permetia la realització d'una plantilla per a Publii des de zero (hagués estat un projecte en sí mateix) es va decidir cercar alguna plantilla que, tot i ser diferent, pogués heretar algunes de les característiques de l'anterior.

Com es va explicar a 10.4, s'ha escollit la plantilla Square que introduïa la interessant possibilitat de posar una imatge gran a la part esquerra, quedant el costat dret per a contingut. És en aquesta part en la que **s'ha conservat l'estructura essencial de l'antiga pàgina inicial**, tenint cadascun dels subapartats de les seccions en forma horitzontal.

D'aquesta manera es va obtenir un primer prototip adaptat a la plantilla Square que finalment no aprofitava cap dels elements dels primers prototips que s'adaptava molt millor a la tecnologia utilitzada.

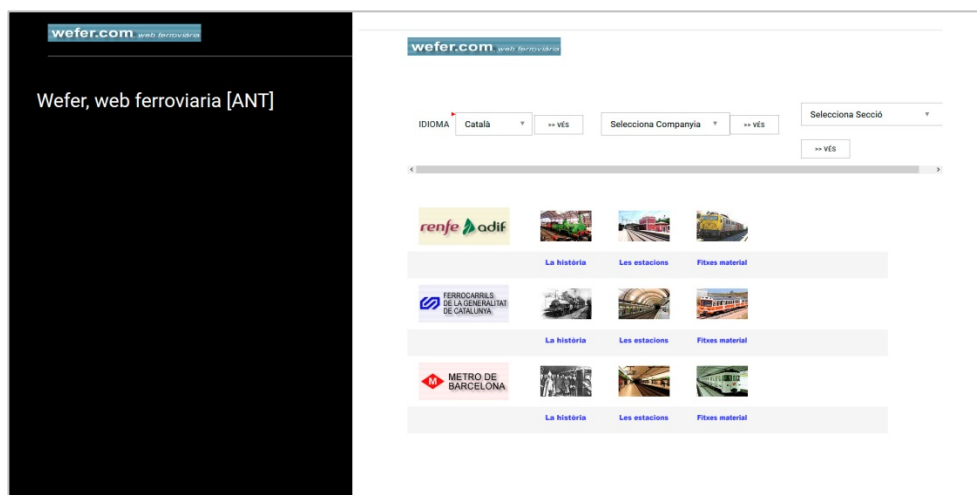


Fig. 14: un dels primers prototips fent servir la plantilla Square

Amb l'evolució de les diferents proves es va veure que amb la configuració utilitzada no s'aprofitava suficientment l'espai vertical i en canvi **es carregava en excés l'espai horitzontal** de tal manera que, amb resolucions tipus pantalla PC de pantalla gran (23" o més) la configuració funcionava correctament, però a la que es baixava la resolució disponible, sense arribar al punt d'haver d'entrar a la versió mòbil, hi havia una evident falta d'espai horitzontal que feia les imatges massa petites o que alguns elements comencessin a muntar.

Per tot plegat es va decidir **verticalitzar una mica més el disseny** passant els logotips de les companyies a una barra horitzontal a sobre dels accessos als apartats, guanyant un 25% d'espai per a les imatges dels apartats i prevenint problemes de responsivitat.

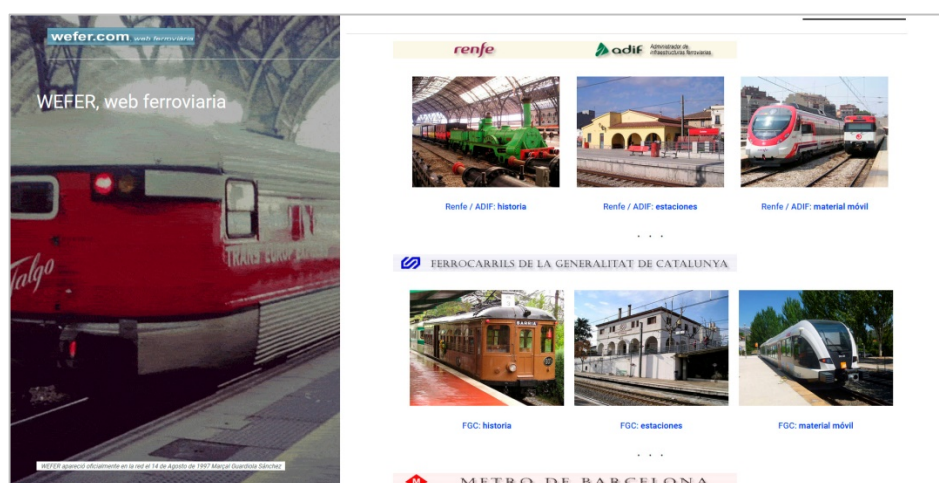


Fig 15: aspecte pròxim al definitiu de la pàgina inicial

13. Perfils d'usuari

Els usuaris que faran servir aquest lloc web són aquells que estiguin interessats en conèixer la història del transport ferroviari a Catalunya. Dins d'aquest perfil primari podem trobar diversos subgrups d'usuaris:

- **Afeccionats al ferrocarril:** són els destinataris principals del web. Existeix a Catalunya una comunitat no gaire gran però sí molt activa (associacions arreu del país, trobades, etc) de persones que tenen com a hobby tots aquells aspectes relacionats amb el ferrocarril i que acostumen a cercar grans quantitats d'informació sobre aquest tema. Aquestes persones estaran interessades sobretot en aspectes com les fotografies d'unitats ferroviàries i també en l'apartat de maquetisme. Arribaran tant a través de cerques com a través de recomanacions en altres pàgines i xarxes socials.
- **Interessats en la història:** moltes persones s'interessen per la història del patrimoni, ja sigui de Catalunya en general o del seu poble en particular. Aquests usuaris arribaran a la pàgina dirigits per una cerca a Google o similar en la qual posaran el nom de la seva població o algun aspecte històric concret. S'interessaran primerament pels textos disponibles i en menor mesura per les col·leccions fotogràfiques
- **Públic general:** són totes aquelles persones que no estan incloses en cap dels dos grups anteriors. Possiblement arribaran a la pàgina a través de cerques directes a Google o similar, ja sigui per paraules o a través de les imatges. Previsiblement acabaran interessant-se principalment per les seccions gràfiques més que pel text ja que majoritàriament no estaran cercant un contingut específic.

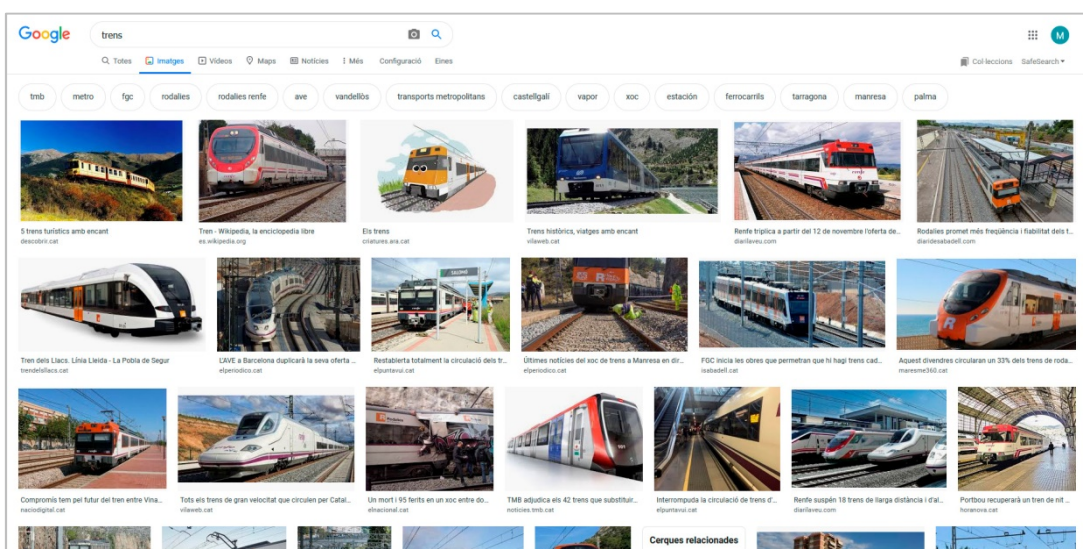


Fig. 16: exemple de la cerca "trens" a Google imatges, una de les possibles entrades d'usuaris al lloc web

14. Usabilitat i experiència d'usuari

Durant el procés de disseny s'ha tingut en compte l'aspecte de la usabilitat i l'accessibilitat com un element principal en el desenvolupament.

Tant l'accessibilitat com l'usabilitat són aspectes essencials per:

- Aconseguir una bona experiència d'usuari de manera que **l'usuari es quedi** en el nostre lloc web en comptes de marxar a altres llocs més "còmodes".
- Aconseguir un **bon posicionament** en cercadors ja que les aranyes que visiten el nostre lloc web donen una mala puntuació si els aspectes d'usabilitat estan deixats. En el cas que el nostre disseny sigui un caos (formalment o a nivell de codi) la puntuació del nostre lloc serà molt baixa o fins i tot no serà correctament interpretat pels cercadors, de manera que no apareixerà a les cerques dels usuaris i no obtindrem visites
- Tenir **bones pràctiques**. Usabilitat i sobretot accessibilitat són essencials pel bon funcionament dels lectors de pantalla de persones amb problemes de visió i cal que ens assegurem, en la mesura del possible, que no hi hagi discriminació.

Per assegurar-se de que l'accessibilitat és correcta, a més de les proves d'usuari és important passar tests automàtics disponibles a la web com per exemple **WAVE (Web Accessibility Evaluation Tool)**:

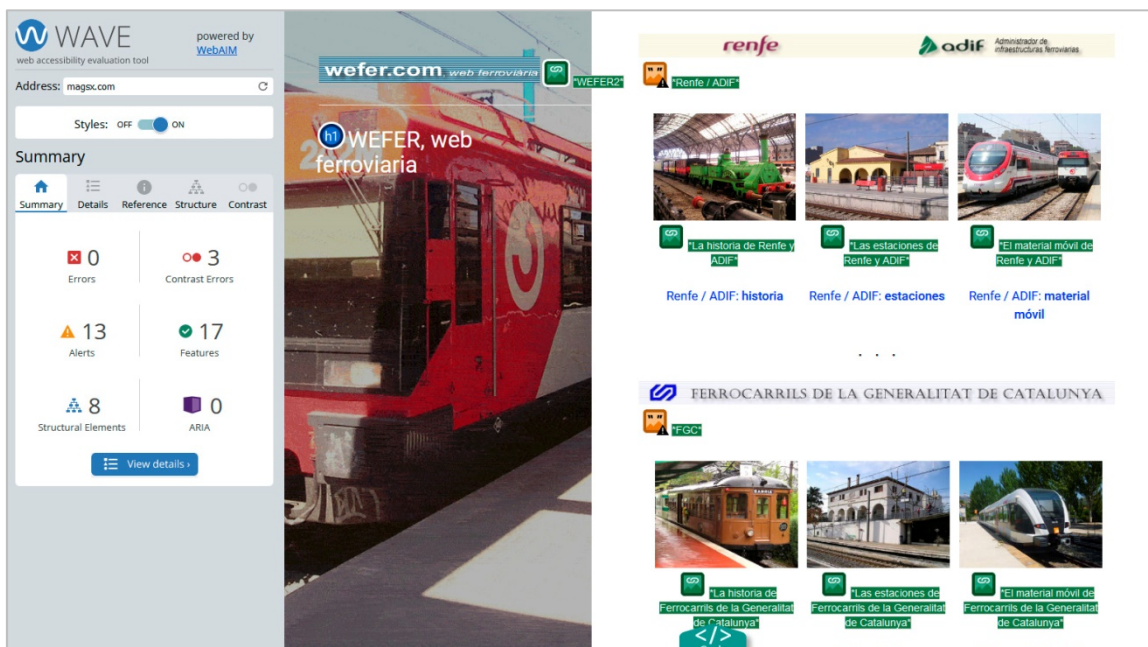


Fig. 17: test d'accessibilitat WAVE on es pot comprovar que la pàgina inicial, ja des dels primers moments de desenvolupament, obté 0 errors principals.

15. Seguretat

Un dels punts forts de la realització d'un lloc **web estàtic** és precisament la **seguretat intrínseca del sistema** respecte als sistemes de CMS dinàmics.

Bona part dels problemes de hacking actualment utilitza atacs tipus **injecció de codi SQL** o similars. En el cas del codi estàtic això és impossible de realitzar.

D'altra banda el fet de **tampoc utilitzar llenguatges dinàmics com PHP** permet desactivar-los per al nostre lloc web des de la configuració del servidor, eliminant un altre dels possibles forats pels quals poden entrar atacants.

Tot plegat és possible perquè tota la part dinàmica, tota la part que pot presentar els problemes més greus de seguretat, **es realitza en el nostre ordinador local**.

D'altra banda aquest lloc web, de moment, **no necessita recopilar dades d'usuaris** per la qual cosa tampoc cal que tingui cap cura especial per haver de desar aquestes dades

La única precaució de seguretat en aquets moments són les **"cookies"**, necessàries pel funcionament d'alguns aspectes interns de les plantilles de Publii. Per complir la llei espanyola en aquest aspecte s'ha inclòs un "pop-up" que ens informa que el lloc web fa servir "cookies" i que aquestes han de ser acceptades en cas que es vulgui continuar navegant pel lloc web.

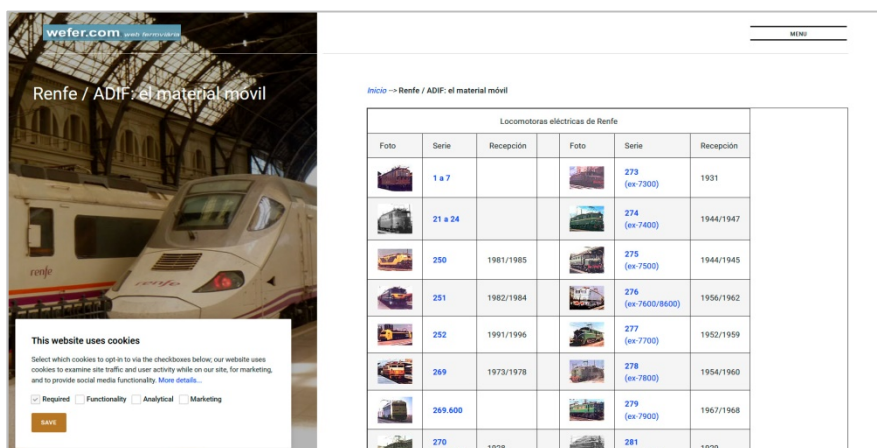


Fig. 18: advertiment de "cookies" en compliment del Reglament General de Protecció de Dades (RGPD)

Una vegada desactivats els llenguatges dinàmics, el punt de seguretat més important seran les connexions FTP. En aquest sentit cal treballar per fer la connexió el més segura possible **utilitzant SFTP** amb una contrasenya forta i assegurar-se que els permisos de les carpetes siguin restrictius, fet al qual ajuda el fet que **el sistema no necessiti escriure res en aquestes carpetes** com sí passa en sistemes que fan servir "caches" per simular pàgines prefabricades.

16. Tests

Els tests s'han realitzat seguint el següent procediment:

S'ha escollit una petita representació d'usuaris representatius que puguin ser usuaris del lloc web i se'ls ha donat una sèrie de tasques a fer,

- Tasques senzilles
- Tasques avançades
- Navegació lliure

Durant la realització d'aquestes proves s'han apuntat els resultats i s'ha preguntat quina és l'experiència que sent cada usuari envers la realització de cadascuna de les tasques.

Amb els resultats, s'han aplicat algunes millores en el funcionament de la interfície, com el canvi dels enllaços o les mides d'algun tipus de lletra.

17. Versions de l'aplicació

Al tractar-se d'un lloc web no podem parlar exactament de versions com ho faríem en una aplicació informàtica, ja que les publicacions poden tenir nous continguts pràcticament cada dia.

No obstant sí **podem parlar de versions del disseny** i de l'estructura del lloc web, que sí són més perdurables en el temps. Podem resumir les versions realitzades fins aquest moment (juny de 2020) de la següent manera:

- **Versió Pre-Alpha (PAC1):** en aquesta versió es va plantejar el primer ús de Publii, explorant les possibilitats de fer servir aquest sistema o qualsevol altre. encara no es tenia clar si es faria servir un sistema estàtic.
- **Versió Alfa (PAC2):** en aquesta versió l'ús de Publii ja estava decidit, amb la qual cosa es desenvolupa l'estructura bàsica del lloc web.
- **Versió Beta (PAC3):** en aquesta versió es treballa sobre el coi CSS personalitzat i s'omple el contingut principal.
- **Versió Producció (Entrega final):** es poleix el codi CSS i s'incorpora una gran quantitat de contingut definitiu. Donades les grans dimensions del lloc web en la seva versió finalitzada, l'entrega cobreix un 20% del que haurà de ser el lloc web definitiu ideal, complint els objectius marcats a l'inici. Es tracta d'un lloc web plenament operatiu disponible per a ser publicat, amb la seva estructura finalitzada i un 20% del contingut final ja disponible.
- **Versions a partir d'ara (Actualitzacions):** a partir d'aquest es continuaran creant les pàgines concretes de cada element, seguint els models ja realitzats, fins completar el total de contingut que s'espera que sigui aproximadament d'unes 500 pàgines i més de 10.000 fotografies.

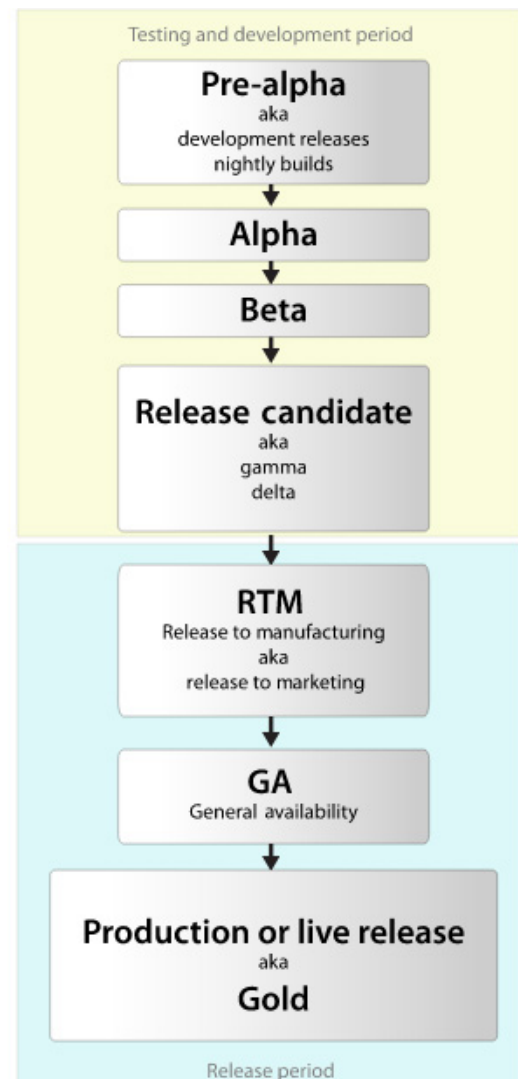


Fig. 19: exemple de control de versions de software. Font: wikipedia

18. Requisits d'instal·lació

El procés ha començat amb la descàrrega de Publii des de la pàgina web oficial:

<https://getpublii.com/>

Segons paraules els creadors del programa, "Publii és un nou CMS estàtic amb una interfície gràfica que permet crear un lloc web HTML extremadament segur, ràpid i elegant. Publii és una eina de creació de llocs web estàtics per a tots els nivells d'usuari, des de principiants a desenvolupadors".

Algunes de les característiques més destacables d'aquest sistema són:

- **Més segur** que la resta de CMS gràcies al fet de fer l'edició en local i no haver de fer servis bases de dades ni actualitzacions de seguretat en el servidor.
- Molt **més ràpid** que la resta de CMS pel fet que el servidor lliura les pàgines sense haver-les de processar cada cop que es demanen ja que aquest pas el fa el creador en local una sola vegada.
- **Personalitzable**. A l'igual que la resta de CMS permet la personalització mitjançant plantilles, edició de codi personalitzat, etc.
- Permet el **treball en equip** ja que encara que el programa estigui instal·lat en local, permet que el codi estigui en serveis al núvol com Dropbox.
- **Simplicitat**: l'editor és visual, té només les opcions essencials, la interfície és senzilla i clara, etc i els autors es comprometen a mantenir-ho d'aquesta manera.
- Programa **gratuït** de codi lliure i que vol continuar sent-ho en el futur.

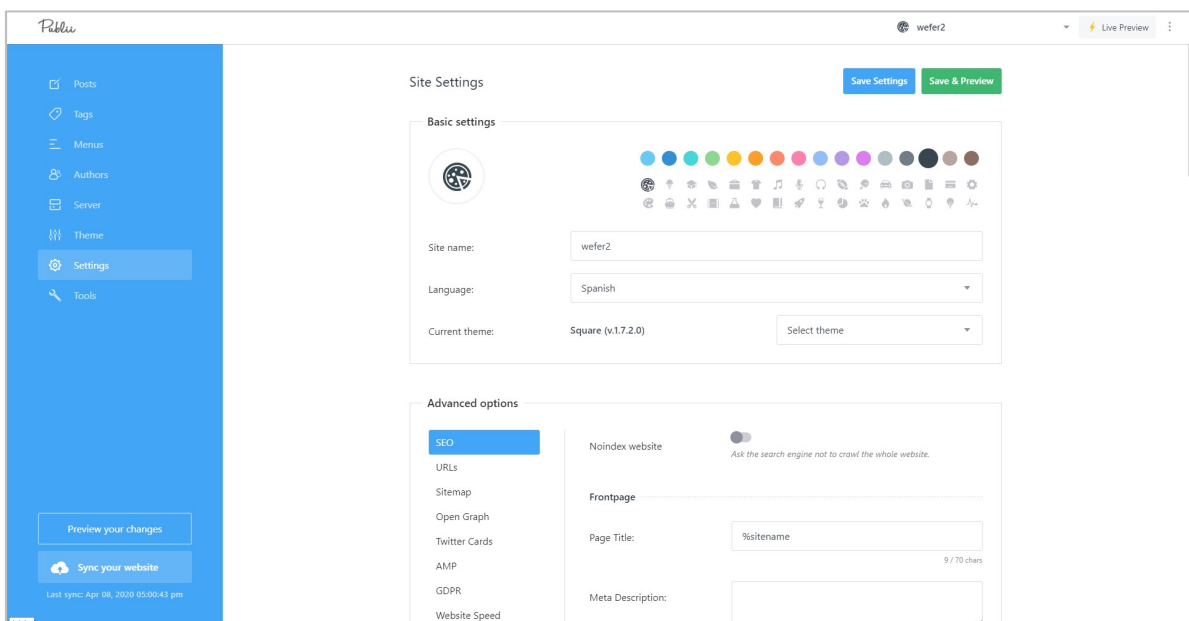


Fig 20: pantalla inicial de Publii

Una vegada instal·lat el programa amb l'assistent habitual (Windows), s'ha executat el programa i s'ha procedit a crear el lloc web a desenvolupar.

Quan entrem a Publii ens trobem amb una barra lateral a l'esquerra que conté els principals apartats de que disposa el programa. A la dreta ens aniran apareixent les diferents opcions que podem canviar i personalitzar i, en alguns casos, una sub-barra lateral que ens donarà accés a d'altres subapartats.

Igualment a la part superior dreta de la interfície hi trobem, el selector per canviar el web que estem editant, a la seva dreta el botó "**Live Preview**" que ens porta directament al web publicat a internet (que diferirà del local en cas que no haguem fet una sincronització després d'haver fet canvis) i finalment un menú amb opcions com accés a l'ajuda en línia, possibilitat de fer donacions als autors o enviar errors que trobem al programa, a més d'accés directe a la configuració global i als temes.

19. Instruccions d'instal·lació

19.1. Creació del lloc web

El primer apartat en el que ens hem de fixar és el de "Settings". Entre la multitud d'opcions que tenim, cal destacar les següents:

- **Site name:** és el nom principal que tindrà el lloc web i que apareixerà en els títols de les pàgines. dins del programa Publii podem tenir multitud de llocs web diferents amb les seves configuracions diferenciades.
- **Language:** en la versió actual només es contempla la possibilitat de tenir un sol idioma. En el futur, segons els "ToDo" dels desenvolupadors, serà possible crear llocs web multiidioma. En el nostre cas s'ha escollit "Spanish" com a idioma. Aquest paràmetre afecta a les traduccions de les plantilles, en cas de que existeixin.
- **Current Theme:** aquí hem de seleccionar el tema que voldrem fer servir. Veure apartat 10.3. per una explicació més àmplia.

En la part inferior, tenim les "Advanced Options". D'aquestes és important modificar:

- **GDPR:** activa l'avís d'utilització de "cookies" per tal de complir la normativa europea de protecció de dades. És important activar-lo per evitar problemes legals.

La resta de paràmetres no són importants per al funcionament bàsic, tot i que podem anar jugant amb algunes d'elles, com per exemple les opcions per canviar el format de les URL, la manera de classificar els posts per defecte, el format dels fils RSS, etc.

19.2. Configuració del servidor

A la barra de l'esquerra tenim l'apartat Server. Des d'aquí podem introduir les dades perquè Publii es connecti al nostre servidor web i pugi els canvis que haguem fet.

Una vegada omplertes les dades hem de comprovar la connexió amb el botó "Test connection"

En el cas d'aquest treball ha sigut impossible mantenir una connexió SFTP ja que dona un error sense especificar. Probablement es tracta d'algun problema de certificats del propi servidor que caldrà estudiar més endavant. Per això i puntualment s'ha decidit continuar el desenvolupament utilitzant FTP convencional, menys recomanable en ser una connexió menys segura.

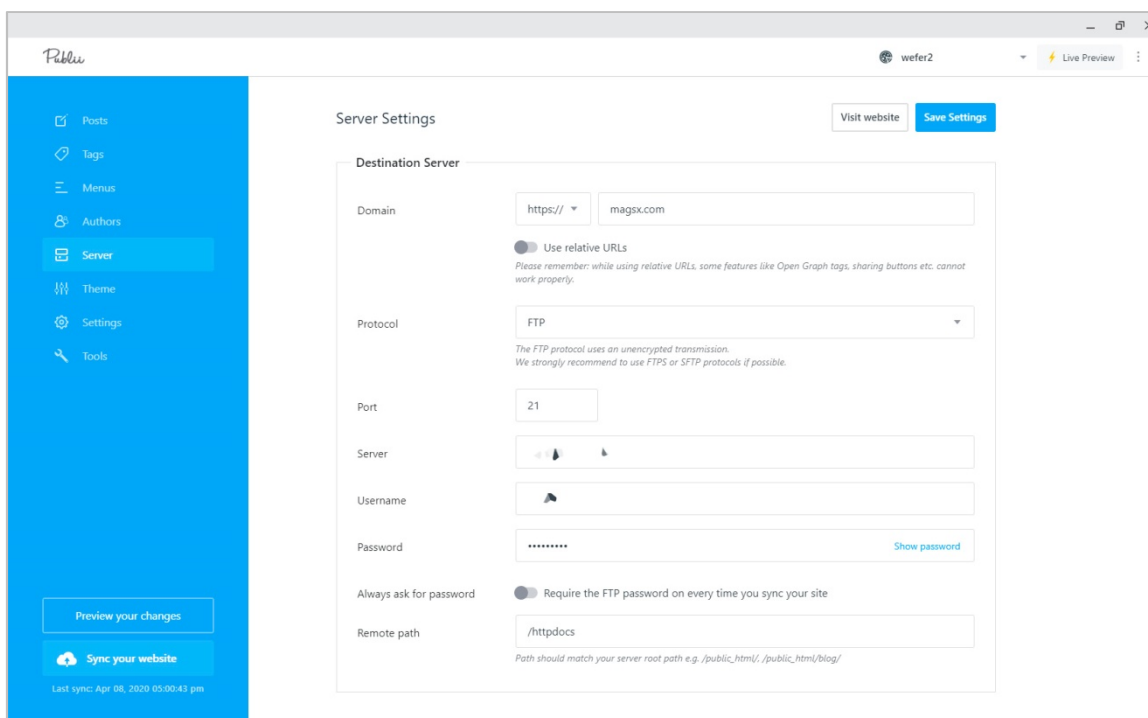


Fig. 21: aspecte de la finestra de configuració del servidor on publicarem el nostre web

19.3. Personalització: temes

Publii, com la majoria de CMS, **és personalitzable mitjançant temes (o skins)** que doten de diferents aparences la nostra pàgina web. En aquest cas el catàleg de temes disponibles és reduït, i més encara si els volem gratuïts.

Des de la pàgina web oficial i des de l'apartat "**Marketplace**" podem descarregar-nos fins a 6 temes gratuïts que, sumats al tema per defecte, totalitzen set temes disponibles. Tot i no ser una gran oferta en número, ens dóna una certa possibilitat d'escollir.

Existeixen altres temes de pagament que aporten més qualitat a la visualització i que sumen suport professional per instal·lar-los i modificar-los per part de l'usuari.

Publii, com els altres CMS, disposa d'una documentació completa per tal que els desenvolupadors puguin crear plantilles per aquest sistema. Podem trobar tota la informació tècnica de com fer aquestes plantilles, com programar-les i quines són les variables disponibles a la següent adreça: <https://getpublii.com/dev/>

Pel nostre projecte el primer pas ha estat veure quina de les plantilles podia adaptar-se millor al projecte desitjat. Donada la magnitud del projecte i el repte tècnic que suposa en sí mateix el disseny amb Publii s'ha descartat poder fer un tema des de zero però no adaptar-lo a les necessitats.

Tal com hem comentat abans, ha calgut anar al "Marketplace" de Publii on estan totes les plantilles disponibles i cercar les que són gratuïtes ja que de moment volem únicament provar les possibilitats de disseny, deixant plantilles professionals avançades per un hipotètic segon pas més endavant.

Per instal·lar una plantilla cal descarregar-la en format .ZIP i descomprimir-la al directori de dades del programa. Aquest directori acostuma a ser una carpeta anomenada Publii dintre de "Documents" de Windows, concretament hem d'anar a:

Equip -> Documents -> Publii -> Themes

Descomprimim cadascuna de les carpetes de cada tema i reiniciem Publii. Ara, ja les tenim disponibles a *Settings -> Current theme*, encara que caldrà clicar a "Install & use" per fer-les servir com a tema pel web que estiguem editant.

Després de fer proves inicials amb tots els temes, es decideix que tres d'ells són els millors candidats per desenvolupar el nou web, concretament els temes anomenats Simple, Technews i Square que presenten aquest aspecte per defecte:

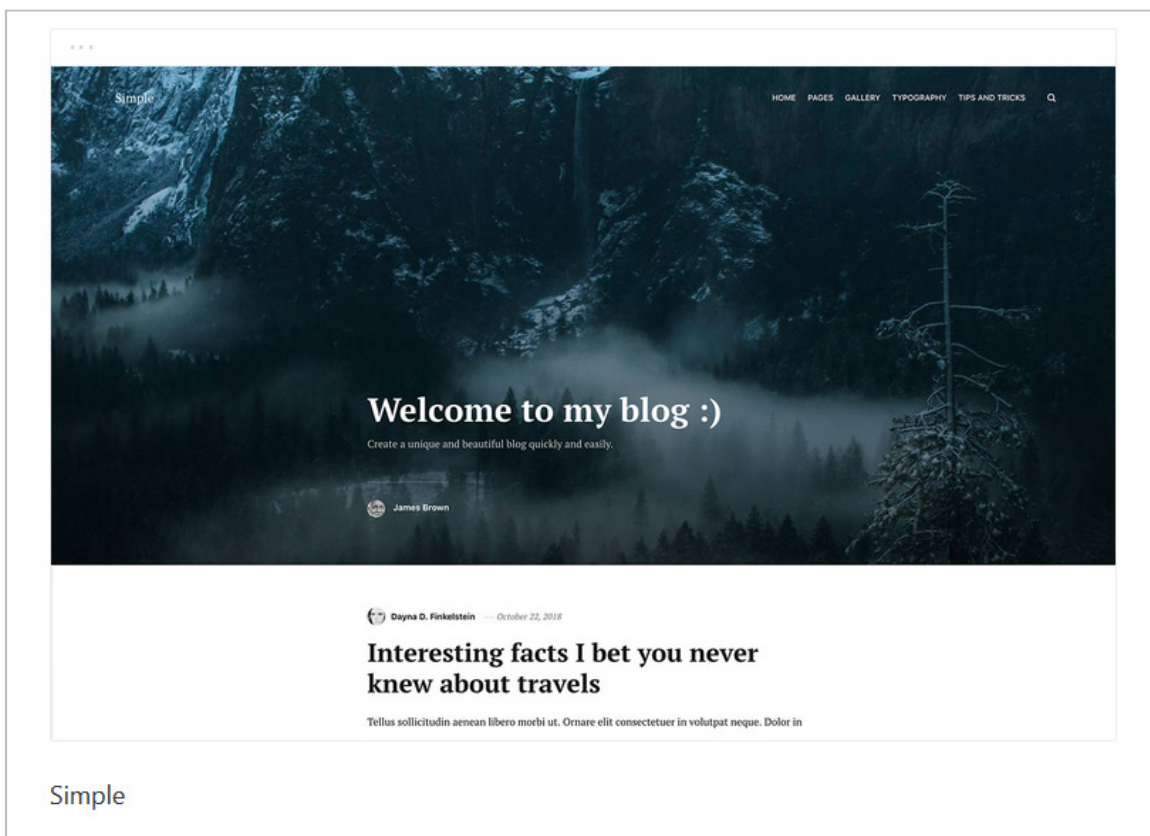


Fig 22: plantilla SIMPLE per a Publii

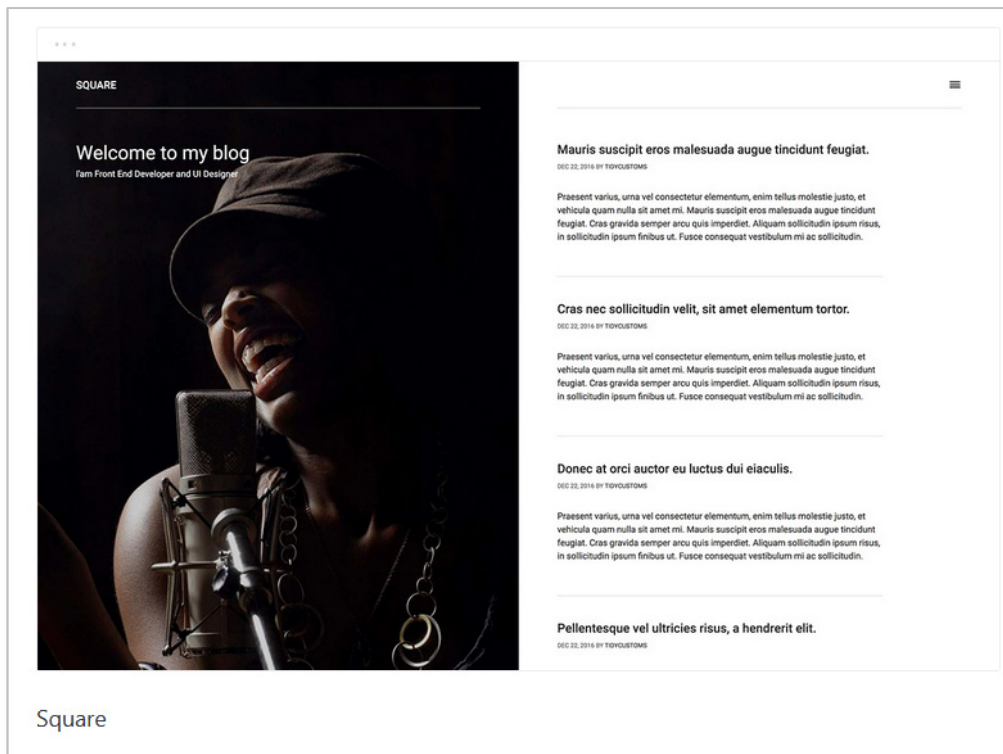


Fig. 23: plantilla SQUARE per a Publii

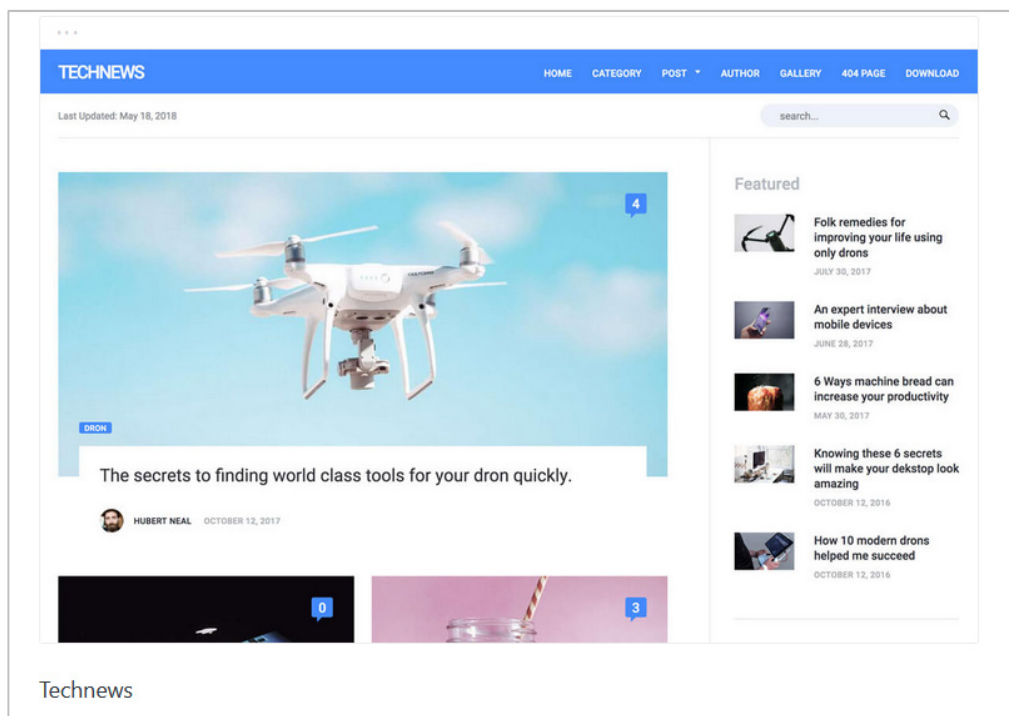


Fig. 24: plantilla TECHNEWS per a Publii

Les tres plantilles tenen en comú una imatge de grans dimensions a la pantalla inicial, possibilitat de presentar a la mateixa pantalla un o diversos posts i un menú per accedir a altres apartats.

Després de diverses proves es decideix que el tema Simple dóna una importància excessiva a la imatge mentre que **la plantilla anomenada *Technews* és més adequada** per un lloc d'informació periòdica i el seu disseny resulta potser massa funcional pel que es busca en aquest cas.

Per tant, s'escull la plantilla *Square* com a plantilla que s'ajusta més a les pretensions d'aquest lloc web, que donarà una gran importància a la fotografia però acompanyada de textos explicatius amplis. Una pantalla dividida en dos com la que presenta aquesta plantilla s'adapta perfectament a aquest requisits.

Una vegada decidida la plantilla a utilitzar, passem a fer l'adaptació definitiva

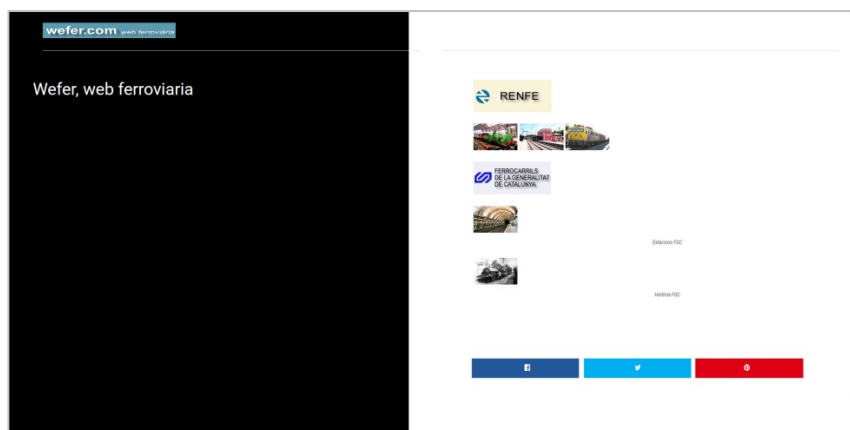


Fig. 25: començament de l'adaptació de la plantilla Square

El primer que es detecta és que l'espai per la imatge lateral esquerra ocupa una part molt gran de la pantalla i, al ser una imatge estàtica, és un espai desproporcionat per l'espai que ocupa. Per corregir-ho es prenen dues vies:

CSS personalitzat: Publii permet escriure CSS personalitzat que sobreescriu el CSS contingut per defecte a cadascuna de les plantilles. Per poder-ho fer primer cal cercar quin és l'element a modificar. Per aquesta tasca utilitzarem les "*Firefox Developer Tools*" que apareixen prement la tecla F12 mentre s'està navegant per qualsevol pàgina amb aquest navegador.

Les "*Firefox Developer Tools*" ens permeten no només veure i navegar pel codi de la pàgina, sinó que resalta a quin element de la pàgina correspon, quines propietats CSS se li estan aplicant i fins i tot ens permet jugar amb el codi i el CSS veient com afecten els canvis que fem a cadascun dels elements en viu.

Això ens permet cercar, per exemple, quins són els elements CSS que determinen l'amplada dels dos contenidors d'aquesta plantilla i canviar-ne les dimensions fins que el resultat sigui el que volem per, tot seguit, anar a l'apartat de **codi CSS personalitzat** de Publi i copiar els canvis fets.

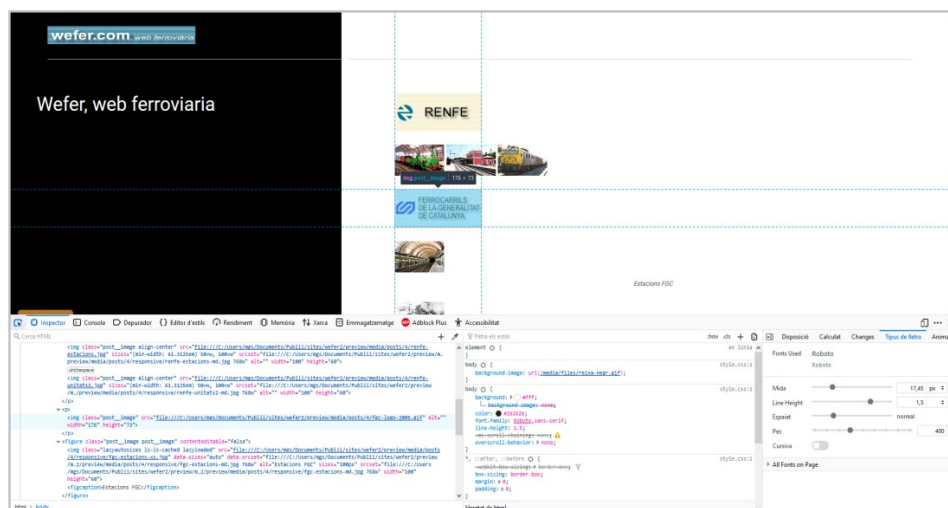


Fig. 26: exploració del codi de la pàgina utilitzant "Firefox Developer Tools"

Una vegada tenim el codi necessari, anem a Publi i accedim, mitjançant la barra lateral, a l'apartat Tools i posteriorment a "Custom CSS". En el quadre de text que ens apareix podem **anar afegint tot el codi que vulguem**. Veurem els canvis quan cliquem sobre el botó "Save & Preview" situat a la cantonada superior dreta.

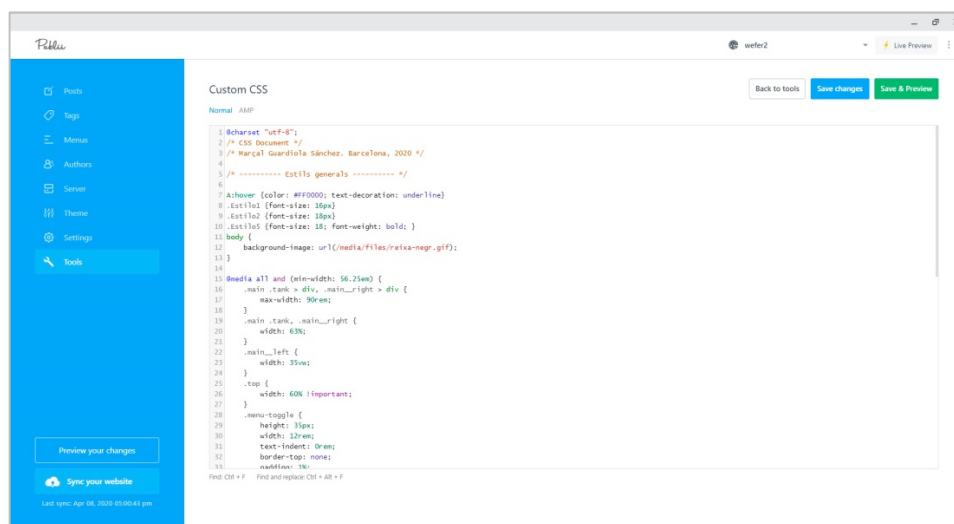


Fig. 27: Codi CSS personalitzat

És important igualment **anar visualitzant els canvis** en el nostre servidor de prova ja que tal com veurem en l'apartat 23. (Bugs) sobretot el CSS personalitzat algunes vegades no es mostra tal com es veurà finalment.

20. Instruccions d'ús (edició)

20.1. Creació d'articles

Per crear un article hem d'anar a la barra lateral esquerra de la finestra principal de Publi i clicar l'opció "Posts", que ens portarà al llistat d'articles:

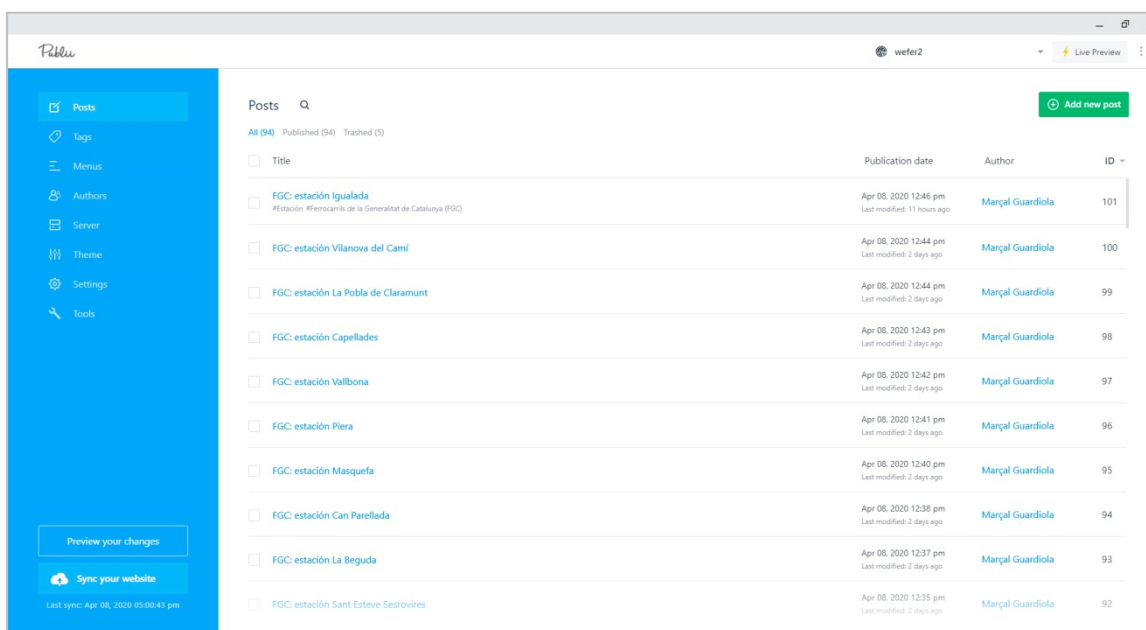


Fig. 28: Llistat d'articles

Des d'aquesta finestra tenim accés a tots els articles que conté el nostre lloc web. Els articles poden ser canviats d'estat des d'aquest llistat. Els articles poden tenir els següents estats:

- **Published:** l'article està publicat normalment al nostre web i és cercable. És l'estat normal dels articles.
- **Draft:** article que està a mig fer. Apareix al llistat però encara no està publicat.
- **Trashed:** article a la paperera de reciclatge. Ja no està al nostre web però es pot recuperar en qualsevol moment a no ser que es buidi la paperera.

A més, poden tenir les següents característiques també editables des d'aquest llistat:

- **Featured:** l'article és prioritari. Normalment apareix en portada si s'escull la configuració automàtica de la mateixa
- **Excluded from homepage:** mai apareixerà a la pàgina inicial, encara que sigui novetat.
- **Hidden:** article publicat però no accessible en les cerques que es facin al lloc web.

A més, des d'aquest llistat també podem duplicar un article i veure directament les etiquetes principals del mateix.

Per **editar el contingut de l'article**, cliquem sobre el seu nom:

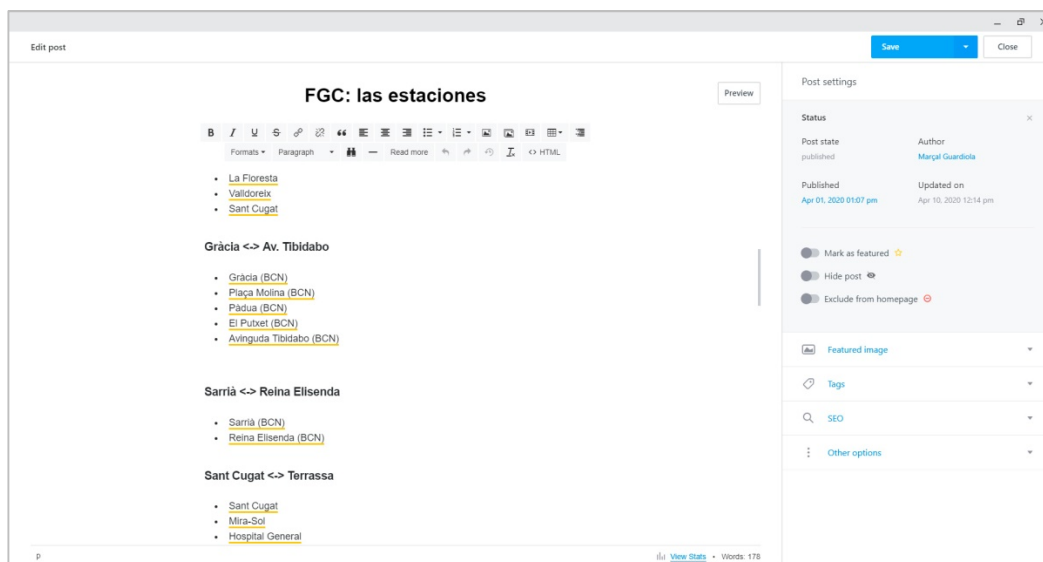


Fig. 29: finestra d'edició d'articles en mode WYSIWYG

A la finestra d'edició tenim els controls típics de qualsevol editor en línia amb l'avantatge que són totalment WYSIWIG. A més podem canviar a editor de codi utilitzant el botó <>HTML:

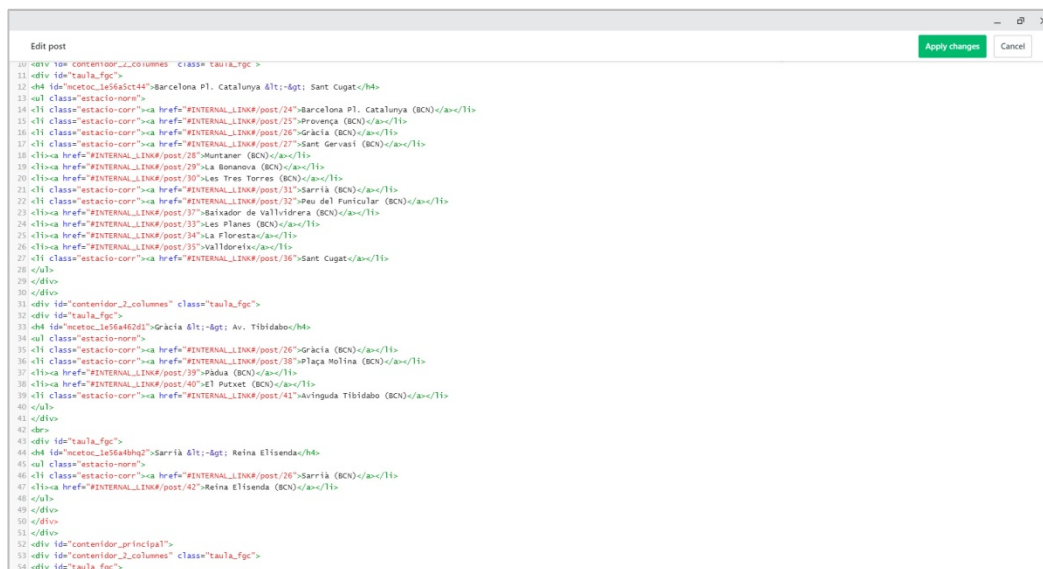


Fig. 30: finestra d'edició d'articles en mode codi HTML

Idealment hauríem d'editar sempre amb l'editor **WYSIWYG**, però en moltes ocasions no funciona correctament (veure apartat 23. Bugs) i a més té carències importants com la creació i gestió de DIVs, entre d'altres. En canvi sí té interessants opcions com la creació automàtica de galeries fotogràfiques, taules, llistes, enllaços i vinculació de vídeos, entre d'altres.

Per fer operacions com afegir DIVs i crear estructures, hem de passar inevitablement per **l'editor de codi HTML**. En aquest editor no tenim botons d'ajuda per col·locar elements, però si tenim una ajuda visual que, teòricament, ens avisa en color vermell si tenim una etiqueta HTML posada incorrectament. Tot i que la fiabilitat no és absoluta acostuma a ser una bona ajuda en cas de problemes.

El codi que veiem és únicament el HTML del contingut de l'article concret. No podem veure encapçalaments, peu i altres elements que no formin part del contingut principal.

Així doncs **la finestra d'edició de codi resulta molt bàsica** i es podria millorar de diverses maneres:

- **No podem veure el resultat de l'edició del codi** fins que no tornem al mode d'edició WYSIWYG, cosa que podem fer en qualsevol moment i tantes vegades com vulguem.
- Si intentem indentar el codi una vegada el salvem aquesta **indentació es perd**, la qual cosa dificulta poder tenir un codi ordenat i clar.
- **No tenim cap botó per afegir codi** directament, per exemple, per fer una llista automàticament o afegir un estil. Sí ho podem fer a la pantalla WYSIWYG però ens cal sortir de l'editor HTML.
- Caldria **millorar l'ajuda visual** de colors perquè ens advertís correctament sempre de les etiquetes que presenten problemes, així com alguna ajuda visual que permetés saber a quina etiqueta d'obertura correspon cadascuna de les de tancament, a l'estil del que fan programes com Dreamweaver.

Precisament per poder editar millor, en algunes ocasions com per editar millor DIVs s'ha optat per editar el codi amb Dreamweaver, que ens ofereix opcions més avançades, i després enganxar-lo directament a l'editor de codi de Publii, amb un resultat plenament satisfactori.

En la part dreta de l'editor tenim els següents controls:

- **Mark as featured / Hide post / Exclude from homepage:** tenen el mateix significat descrit en el llistat d'articles
- **Featured image:** és la imatge principal de l'article. És la que es fa servir en la mostra resum de l'article o, en el cas de la plantilla utilitzada, la que es fa servir per la zona esquerra de la pantalla on es mostra la imatge en gran.

- **Tags:** paraules clau de l'article, Aquestes paraules permeten trobar millor els articles quan es fa una cerca. Permet un número il·limitats d'etiquetes però només una d'elles es pot posar com a principal. Addicionalment, com que Publii no té un sistema de jerarquització de l'estructura mitjançant categories, caldrà que fem servir aquestes paraules clau per ordenar els nostres articles per temes. Les etiquetes es poden gestionar des d'una pàgina específica accessible des de la barra principal de la pàgina principal de Publii.
- **SEO:** opcions per optimitzar la manera com els robots de cerca indexaran la pàgina, tals com text de la meta descripció o personalització del títol per a robots.
- **Other options:** opcions diferents de visualització. La majoria es configuren per defecte des de la configuració principal de Publii, com són la visualització del nom de l'autor de l'article, la data de creació, etc.

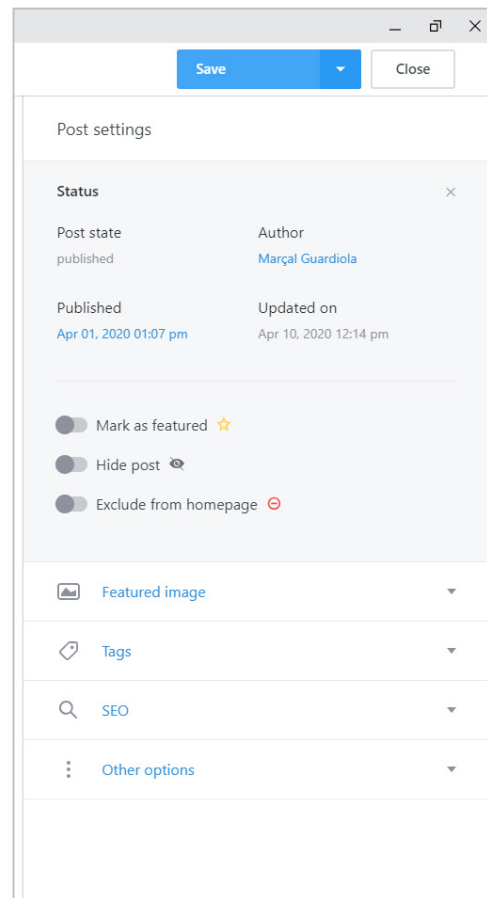


Fig. 31: barra lateral d'edició d'articles

Com a últim apartat i com a curiositat, a la part inferior de l'editor tenim un enllaç a les estadístiques de l'article que ens mostren:

- Quantitat de paraules de l'article
- Número de paraules úniques
- Número de caràcters totals
- Quantitat de frases
- Número de paràgrafs
- Temps de lectura aproximat

En resum, per l'edició normal dels articles Publii és més que suficient i únicament en casos en que vulguem anar una mica més enllà de l'habitual necessitarem l'ajuda d'un programa extern com Dreamweaver.

Si en el futur es millora el sistema d'edició permetent elements més avançats com dibuixar DIVs i la indentació del codi, aquest editor serà més que suficient per qualsevol ús que vulguem donar-li als nostres articles.

20.2. Inclusió d'imatges

Paral·lelament a la creació d'articles principals en quant a text i estructura, s'ha procedit a **crear i retocar les imatges i logos** que s'han de fer servir en el nou lloc web.

Per als logos s'ha fet servir de base els logotips que ja existien a l'antiga web ja que, tot i tenir una estètica millorable, ajuden a donar continuïtat formal a l'antic web i serveixen com a record del mateix.

Per l'edició dels logotips s'ha utilitzat **Adobe Photoshop**. També s'ha utilitzat aquest programa per editar imatges més complexes com les dels plànols, fent servir el treball per capes i objectes possible en aquest programa.

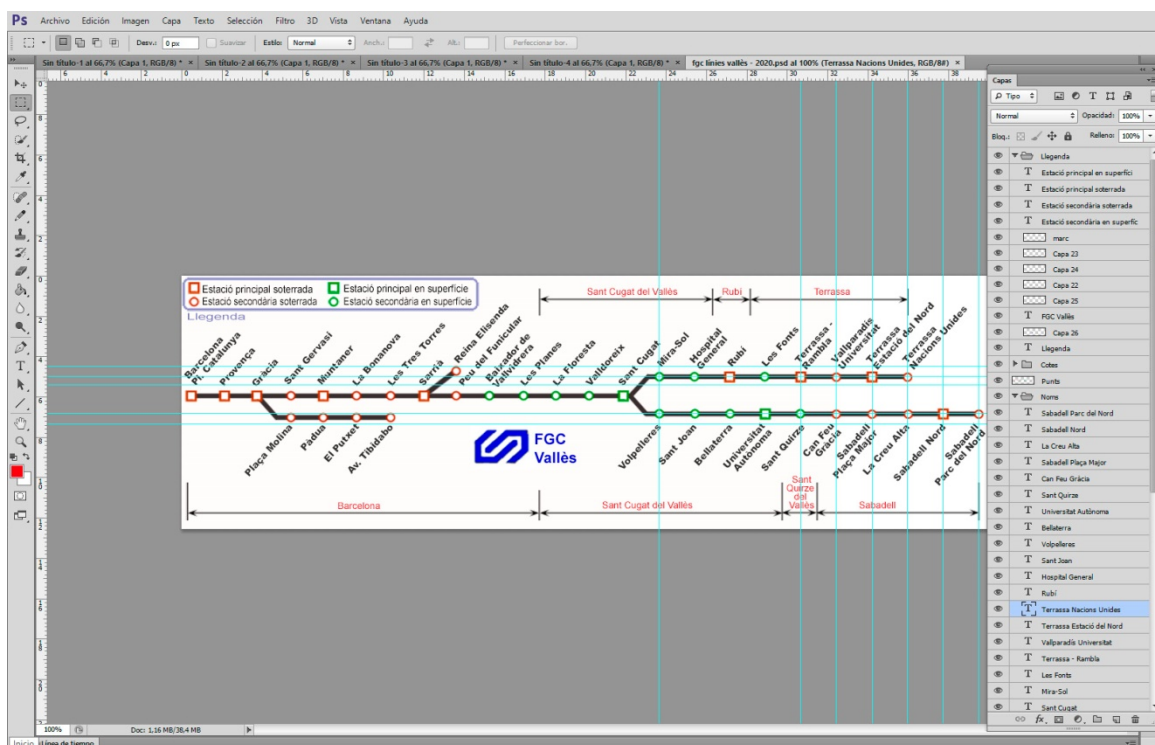


Fig. 32: procés de treball amb Photoshop

La majoria de **logos** s'han desat en **GIF o PNG**, segons s'ha vist que el resultat és millor en un o altre format (mida/qualitat), mentre que les **fotografies** són **JPG**.

Totes les imatges s'han optimitzat per la seva publicació a internet utilitzant compressions mitjanes i resolucions adaptades a la web. Igualment hem de tenir en compte que Publii genera automàticament miniatures (thumbnails) de les imatges que pugem per adaptar-les en cada moment a la resolució més adequada segons sigui necessari.

A més, el disseny és totalment **responsive** per la qual cosa no hem de preocupar-nos per la mida en cadascuna de les resolucions possibles.

Un cop tenim els logos o les imatges preparats, per incloure'ls en un dels nostres articles tenim dues possibilitats:

- **Incloure'ls directament des d'un directori del nostre ordinador:** seleccionem la imatge des de la carpeta on tinguem la imatge i Publii importa directament la imatge. Pot generar una mica de confusió veure que en el codi HTML Publii enllaça la imatge com un fitxer local:

file:///C:/Users/usuari/Documents/Publii/wefer/input/media/posts/11/imatge.gif

No obstant aquesta adreça és només utilitzada pel programa per carregar la imatge, ja que durant el renderitzat es substitueix per l'adreça del nostre lloc web.

- **Pujar la imatge a la galeria multimèdia de Publii:** cada lloc web té un espai multimèdia propi dins del programa, a forma de contenidor on podem desar imatges i altres arxius multimèdia. No obstant no ofereix un avantatge clar respecte a carregar-les directament en la forma anteriorment descrita i, a més, si es troben en a la galeria multimèdia són més difícils d'incloure en un article ja que s'ha de posar l'adreça a mà al no tenir opció d'accedir a la galeria de cap manera. Per tant, és un aspecte a millorar o eliminar definitivament.

Una de les parts més importants del web seran les col·leccions de fotografies d'un mateix lloc. Amb Publii **podem crear carrusels clicant un sol botó** a través de l'editor d'articles, generant-se automàticament la galeria i els controls.

No obstant tenim el problema de que les imatges poden ser copiades per persones que en vulguin treure algun tipus de profit sense **tenir-ne els drets**. Per això en aquest projecte una de les premisses ha estat protegir les imatges d'alguna manera que dificulti aquest aprofitament il·lícit.

La solució s'ha trobat en el programa **Photomarks per a Windows**, que ens permet agafar una gran quantitat de fotografies i afegir-los efectes com **marques d'aigua** de diversos tipus a la vegada sense haver-les d'editar una per una. Aquest programa presenta evidents avantatges pel que fa a la senzillesa de funcionament, velocitat i possibilitats específiques de transformació per lots respecte a fer la feina amb programes com Photoshop, encara que fem servir accions gravades. L'eficiència i rapidesa del procés és bàsica donat que es preveu que el lloc web tingui més de **10.000 fotografies** quan estigui acabat.

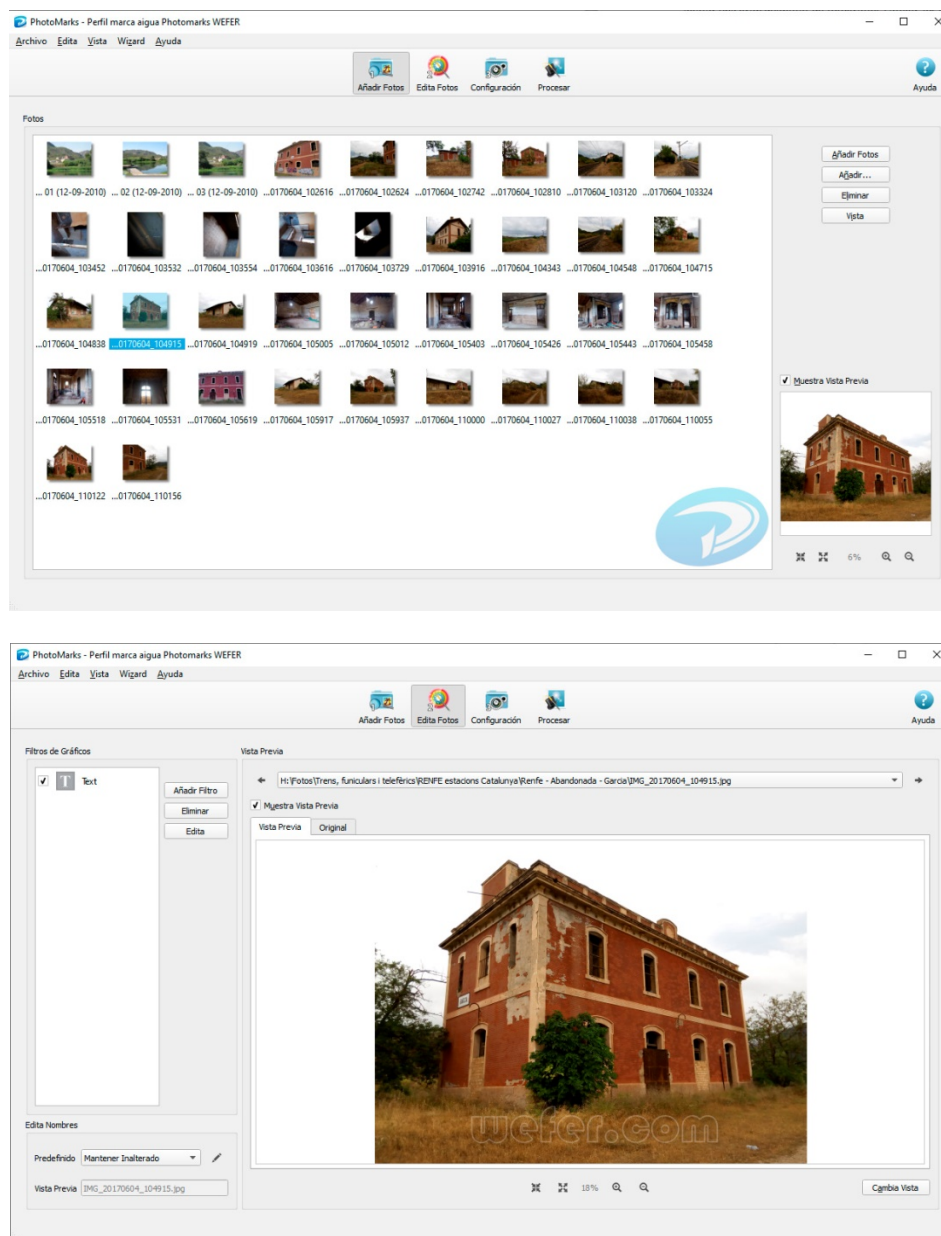


Fig. 33: Inclusió d'una marca d'aigua en un conjunt d'imatges amb PhotoMarks

20.3. Altres funcions

A part de tot l'anteriorment descrit en el desenvolupament del lloc web també s'han hagut de realitzar les següents accions:

- Creació del menú principal:** des de la barra lateral de la pantalla principal tenim accés a la creació e menús. des d'aquí podem crear un menú amb els apartats i enllaços i deixar-lo preparat per ser inclòs en algun lloc prèviament dispostat per la plantilla que estiguem fent servir. en el cas que ens ocupa només existeix la

possibilitat de tenir un menú principal, accessible des d'un botó col·locat a la part superior dreta de cada pàgina del lloc web.

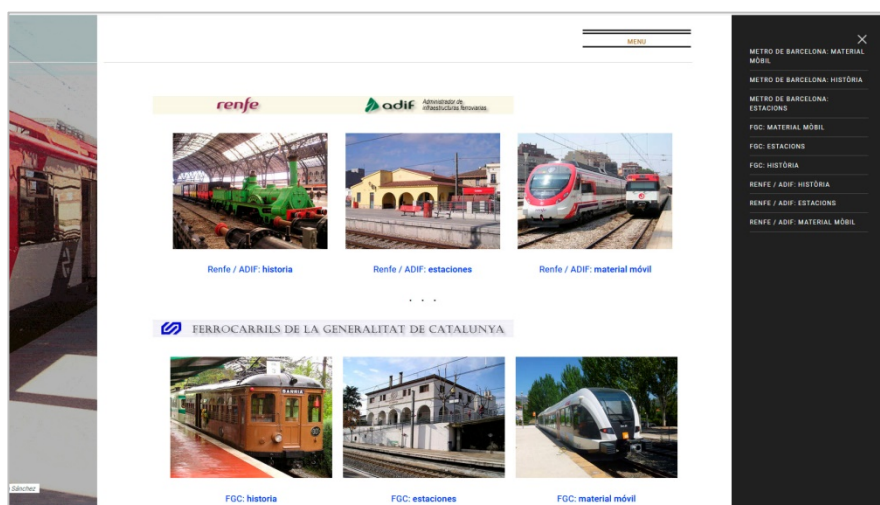


Fig. 34: pantalla inicial amb el menú principal desplegat a la dreta

- Creació de Backups:** des de l'apartat "Tools" de Publii podem accedir a l'opció "backups". Ens apareix una pàgina que ens mostra els backups que tenim del nostre lloc web. Sempre es tracta de backups complets, ja que l'aplicació no permet fer backups incrementals. els backups es guarden en local i no són més que una còpia de tots els arxius del lloc web encapsulats dins d'un **arxiu comprimit tipus .tar**. el llistat ens permet fer un nou backup, esborrar algun/s dels existents o explorar el contingut d'un dels backups, sempre que tinguem alguna aplicació que pugui obrir arxius .tar comprimits. Durant el procés de creació s'han anat realitzant diversos backups i s'han assegurat copiant-los en carpetes externes per tal d'assegurar-se no perdre la feina feta en cas d'error o fallada del programa.

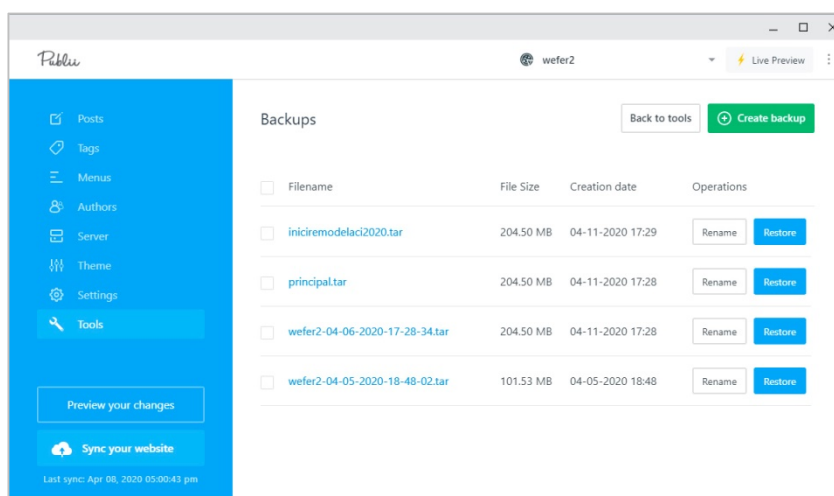


Fig. 35: pantalla de backups

- **Renderitzat i sincronització:** una vegada tenim els nostres arxius editats, les imatges col·locades, l'estructura preparada, etc, ha calgut posar la web en línia. Per això cal fer la sincronització del lloc local amb el lloc en línia a través del botó "Sync your website" de la pàgina principal. Aquest procés automatitzat segueix els següents passos:
 - Publii fa un **renderitzat del lloc**. Això significa que llegeix els arxius, munta les imatges col·loca les adreces adequades perquè funcioni en línia, etc. Per tant, aquest procés converteix les pàgines "en brut" en les **pàgines definitives en HTML** que es carregaran al servidor. Cal fer notar que aquest procés seria equivalent (no igual) a la generació dinàmica que fan els servidors amb els CMS convencionals (combinant dades més codi) cada vegada que es demana una pàgina, però la diferència en aquest sistema de pàgines estàtiques és que el renderitzat es fa únicament quan es puja el lloc web i les pàgines es deixen ja preparades per descarregar en el servidor per part el client, sense haver-se de processar novament del costat servidor.
 - Amb les pàgines definitives preparades, el servidor obre una **sessió FTP** amb el servidor per saber quines pàgines han estat actualitzades respecte a les existents a la còpia antiga que està en el servidor. Això es fa per no haver de pujar cada vegada el contingut sencer del lloc web cada vegada ja que això faria les sincronitzacions eternes.
 - Una vegada el servidor sap quins són els **arxius modificats** (per data) puja els nous arxius no existents i sobreescriu els ja existents fins tenir tot el lloc web actualitzat. Com a crítica cal dir que aquest procés no té en compte aquells arxius que estan en el servidor web però no estan al lloc web local de Publii, de tal manera que aquests arxius antics o que no tenen ja cap relació amb el lloc web no s'esborren fent que, amb el temps, pugui quedar una quantitat important de brossa que l'únic que farà serà ocupar espai.

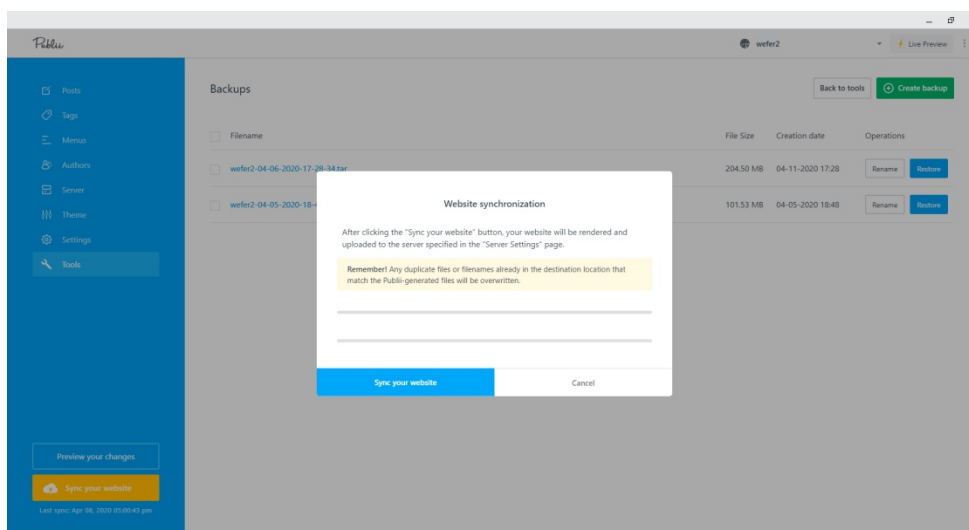


Fig. 36: pantalla de sincronització de Publii

21. Errors ("bugs")

En aquest apartat haurem de diferenciar dos aspectes de la mateixa. En aquest treball, al tractar-se de la realització d'un lloc web estàtic, haurem d'analitzar els dos elements que la componen de manera separada, però íntimament lligada: el **backoffice local** i el **frontoffice** per l'usuari final.

21.1 Backoffice: Publii

Tal com ja s'ha comentat, en els CMS tradicionals el backoffice és una part més del lloc web que es troba en el servidor. Però en el nostre cas no, sinó que aquest backoffice es troba en el nostre ordinador local i està format per una aplicació específica, en aquest cas Publii.

Analitzant Publii, ens trobem que està en la seva versió 0.35.3, essent una versió Beta. Això significa que teòricament encara no és prudent utilitzar-la per llocs en producció. Però després d'haver treballat moltes hores amb ell i haver provat la majoria de les seves opcions s'ha constatat que l'aplicació és totalment estable, fins i tot més que algunes que estan en fase definitiva, i els únics problemes que tindrem podran ser la falta d'algunes opcions importants o petits desajustos que s'han de millorar.

Els principals errors que s'han trobat han estat:

- **Desajustos en la visualització del renderitzat previ.** A vegades aquest renderitzat previ en local no coincideix exactament amb el que es mostra en línia. A la figura següent ho podem veure amb una llista amb les vinyetes personalitzades mitjançant codi CSS personalitzat en el propi editor de Publii:



Fig. 37: diferències de representació entre previsualització local i definitiva remota

Si bé les diferències no són abismals sí que fan que els canvis s'hagin de validar en un servidor de prova abans de ser passats al servidor final, la qual cosa fa perdre un dels principals avantatges que té la forma de treball en local de Publii. No obstant, com que les diferències acostumen a ser petites en principi és recomanable prescindir del servidor de prova i pujar el resultat al servidor definitiu, sempre i quan es comprovi el resultat en aquest.

- **Error de connexió en la sincronització:** tot i que molt estranys, de tant en tant Publii no és capaç de connectar amb el servidor remot per actualitzar, donant un error de connexió. No obstant la solució és tornar a clicar el botó de sincronització.
- **Ralentització del programa:** en Windows 10, si el PC es suspèn en comptes d'apagar-se diverses vegades, el programa experimenta una ralentització molt important que es soluciona simplement tancant el programa i tornant-lo a obrir.
- **Funcionament millorable dels controls:** si carreguem una imatge des del disc dur i cliquem en algun lloc de la pantalla abans de que el programi l'hagi processat totalment, la imatge no es carrega i no ens avisa. Si cliquem el botó "OK" mentre s'està important una galeria d'imatges, la importació s'interromp sense cap notificació, etc.

Més enllà de l'anteriorment apuntat no s'han trobat errors d'importància en l'aplicació **ni s'ha registrat cap sortida inesperada del programa**, tant típiques de la majoria de betes.

Sí que s'han trobat a faltar elements que, sense ser errors, són **faltes de característiques bàsiques que poden derivar en errors**. Per exemple el programa no permet treballar dins dels articles fent servir DIVs, un element bàsic de l'actual web, de manera que els hem d'introduir manualment amb l'editor de codi. Encara així, si després continuem treballant amb l'editor visual ens podem trobar que l'interpret de codi de Publii ens desfà l'estructura de DIVs creant un desastre de codi difícil d'arreglar.

21.2 Frontoffice. Pàgina publicada

S'ha aconseguit una interfície amb un funcionament més que satisfactori en general. L'únic bug important que s'ha detectat i sobre el qual s'ha de continuar treballant és la representació d'alguns elements en **resolucions baixes tipus mòbil**, on malgrat l'adaptació responsive encara alguns elements concrets, com paraules llargues en els títols, poden suposar petits desquadraments que provoquen incoherències estètiques però que no impedeixen l'accés correcte al lloc web.

22. Pressupost

Pressupost pel disseny del lloc web i el manteniment durant el primer any, tot inclòs:

Concepte	Element	Quantitat o proporció d'ús per al projecte	Cost	Cost total 1 any
Maquinari	PC (torre+perifèrics) utilitzats per la creació i gestió del backoffice	0.1	900 €	90 €
Programari	Windows 10 / Adobe Cloud / Photomarks utilitzats pel disseny i manteniment	0.1	500 €	50 €
Serveis	Servidor web per l'allotjament del lloc web pàgina		96 €/any	96 €
Serveis	Registre/manteniment del domini		34 €/any	34 €
Hores personal disseny		120	20 € / hora	2.400 €
Hores personal manteniment i actualització		50	22 € / hora	1.100 €
TOTAL				3.770 €
TOTAL + IVA				4.561,70 €

Taula 02: pressupost

23. Anàlisi de mercat

23.1 Sistemes per a la realització de webs estàtics

En aquest apartat examinarem quins són els diversos competidors de Publii dins del mercat de la publicació de llocs estàtics.

Realment podríem dir que en aquest moment no existeix cap competidor per a Publii, és a dir, cap sistema que sigui 100% igual que Publii. No obstant, tenim molts altres sistemes que podem fer servir per generar llocs web estàtics però que presenten diferències notables respecte a Publii.

Actualment existeixen dotzenes de generadors de llocs web estàtics. Una llista força completa es pot veure aquí: <https://headlesscms.org/>

Alguns dels sistemes més utilitzats són **Netlify CMS**, **Siteleaf**, **Forestry**, **Grav**, etc, però no n'hi ha cap que s'acosti a les característiques de Publii.

El treball amb la majoria de generadors de llocs estàtics actuals implica tenir un coneixement profund de codi. En la majoria de casos la instal·lació és complicada per a persones que no siguin programadors, en alguns casos s'utilitzen fitxers "*markdown*" per editar els arxius en comptes d'editors WYSIWYG, es necessiten tenir coneixements de treball amb versions, etc.

Per la seva banda Publii és extremadament senzill. Per instal·lar-lo només cal baixar-lo i clicar "Endavant" en el seu instal·lador, obrir-lo, configurar el FTP del nostre servidor... i ja el tenim preparat per editar el nostre lloc web.

A més, el treball en local per part del desenvolupador el fa pràcticament únic i molt més proper a l'usuari bàsic, el qual està habituat a treballar amb aplicacions en el seus dispositius.

En resum: per tot plegat, tot i que hi ha molts sistemes que permeten obtenir els mateixos resultats, actualment **en el mercat no hi ha res comparable a Publii**.

23.2. Anàlisi d'un alternativa: Jekyll

23.2.1. Què és Jekyll?

Segons la definició que ens proporcionen els mateixos creadors de l'eina, "Jekyll és un generador de llocs estàtics. Dóna-li un text escrit en el teu llenguatge de marcatge favorit i utilitza dissenys per crear un lloc web estàtic. Pots modificar la visualització de les URL i les dades que es mostren al lloc web i molt més."

La idea de Jekyll és generar el contingut d'un lloc web estàtic a partir d'arxius de marcatge, normalment "markdown" (tot i que podem fer servir HTML, CSS, etc) . Una vegada hem escrit el codi del nostre arxiu el processarem mitjançant jekyll fent servir la línia de comandes del nostre sistema operatiu.

Aquest procés ens genera la pàgina web o lloc web desitjat en llenguatge HTML, el qual després podrem pujar al nostre servidor.

23.2.2. Instal·lació de Jekyll

La instal·lació de Jekyll és força complicada per a un usuari bàsic o mitjà. A més, tota la documentació es troba en anglès per la qual cosa pot haver-hi una barrera d'entrada molt gran per a la majoria d'usuaris. Per això en aquest text s'explica el procés de manera detallada.

Per poder fer funcionar Jekyll, abans hem d'instal·lar altres elements necessaris tal com s'explica en la pàgina web de Jekyll, on ens proporcionen enllaços a tots aquests elements:

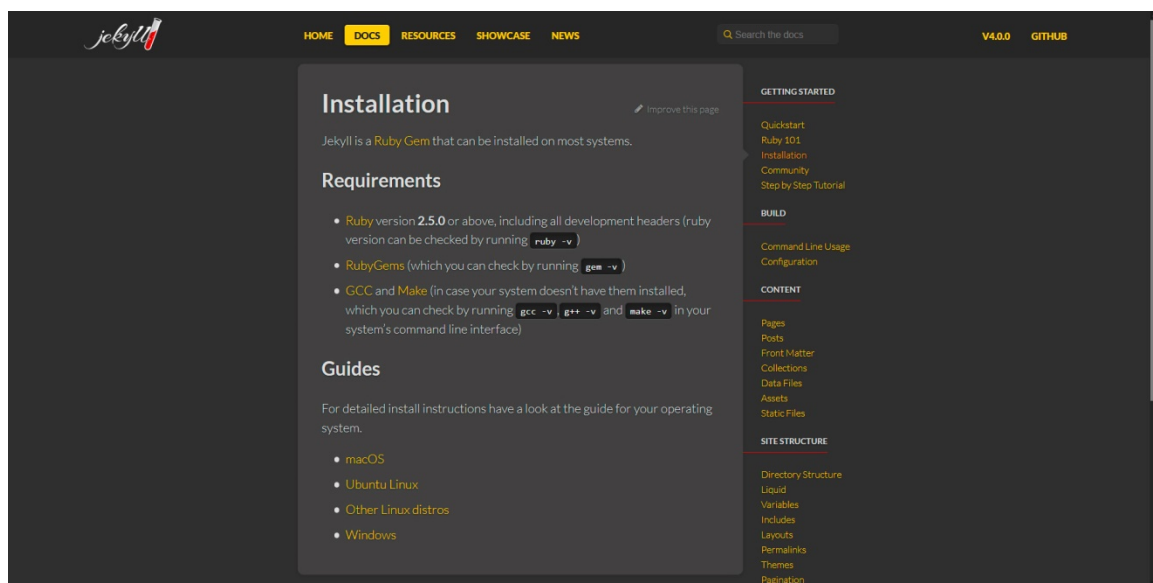


Fig. 38: explicacions de Jekyll sobre els requisits per fer-lo funcionar

En primer lloc instal·lem **Ruby**, un llenguatge de programació orientat a objectes. En alguns sistemes ve instal·lat per defecte, però no és el cas a Windows 10 que és el sistema operatiu en el que realitzarem la instal·lació. Per això és necessari anar a la pàgina web de Ruby i baixar al nostre ordinador l'arxiu adequat per al nostre ordinador:

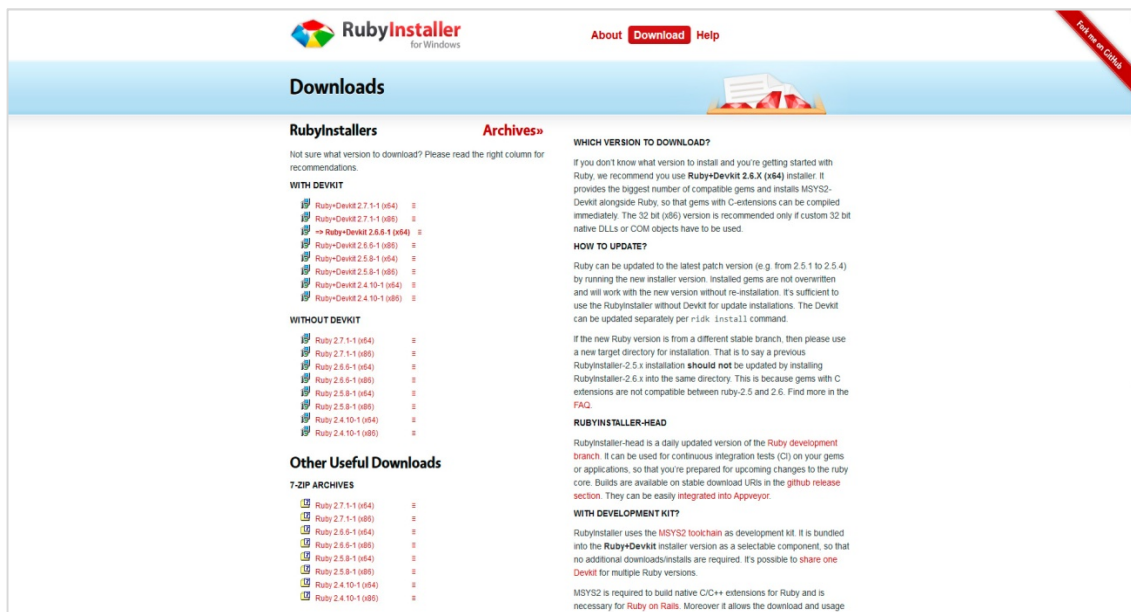


Fig. 39: web de baixada de Ruby: <https://rubyinstaller.org/downloads/>

Tot seguit executem l'arxiu que hem descarregat ja que es tracta d'un executable i seguim un senzill procés amb l'assistent.

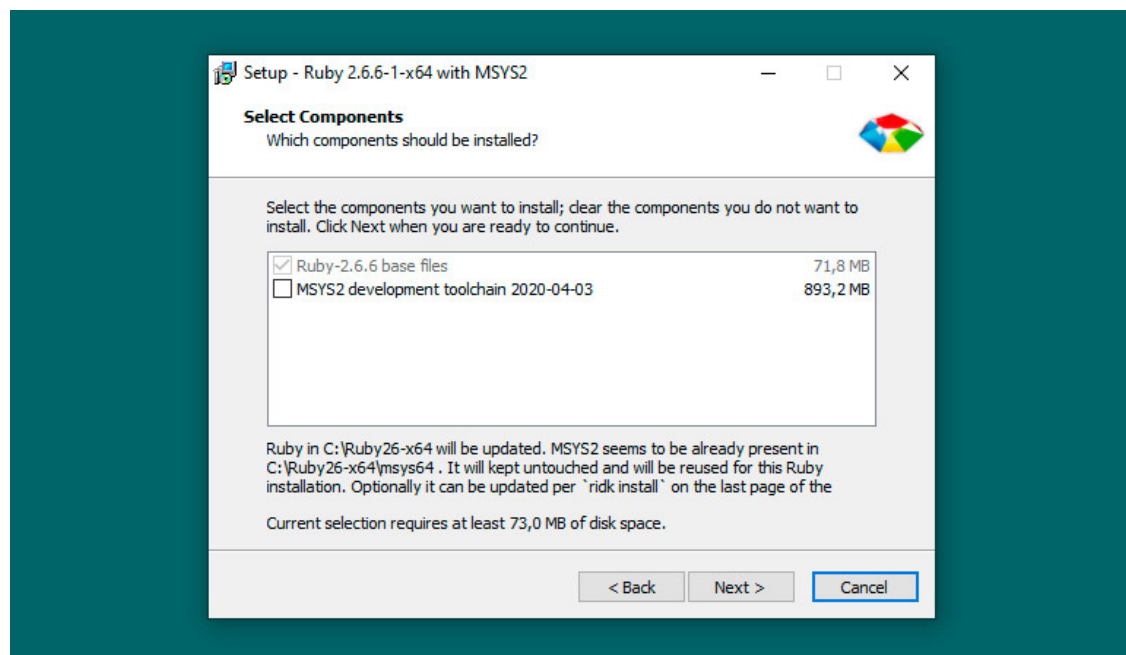


Fig. 40: instal·lació de Ruby a Windows 10

23.2.3. Funcionament de Jekyll

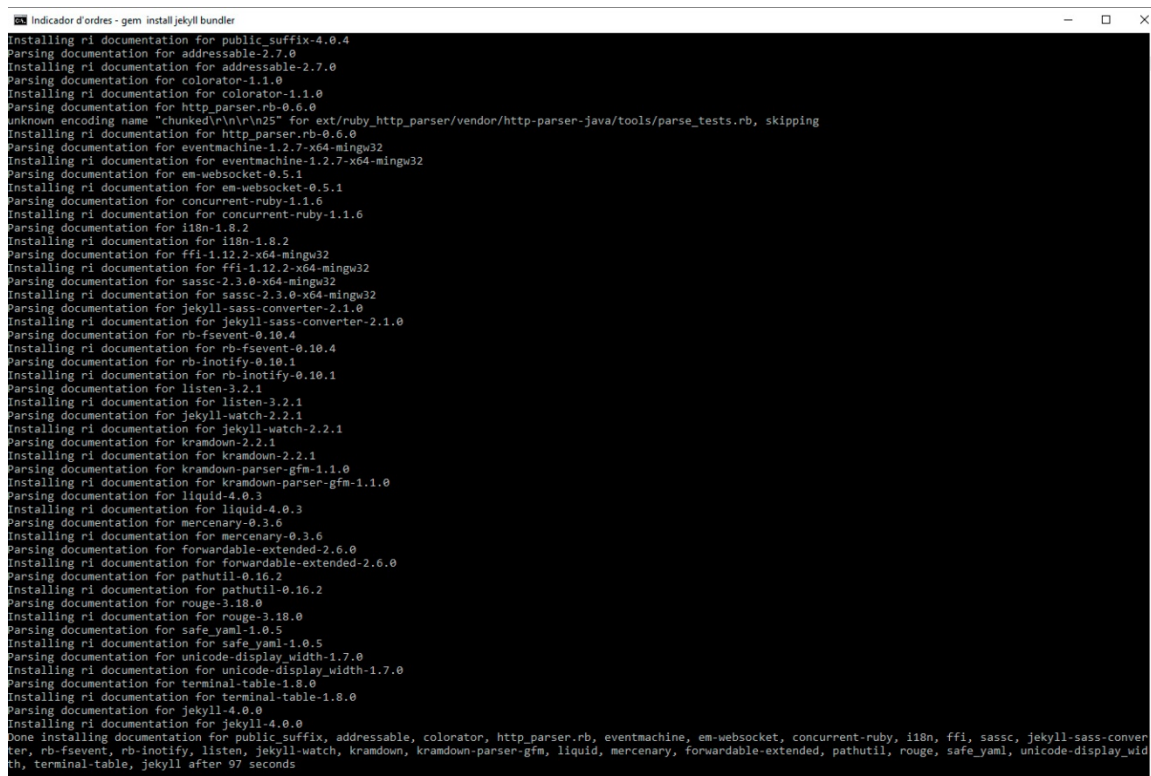
El següent pas consisteix en **executar Ruby** al nostre ordinador.

Per això, anem fins a la consola del nostre Windows (cmd.exe) i executem Ruby. comprovem que també s'han instal·lat **Ruby Gems** i **GCC** (si descarreguem **Ruby + DeveloperKit** ve tot inclòs) i en el cas que no sigui així els instal·lem seguint les instruccions que trobem a la web de Jekyll.

Ara ja podem passar al següent pas que és la instal·lació de Jekyll pròpiament dita. Per això, simplement teclegem el següent text a la consola de Windows:

```
gem install jekyll bundler
```

La qual cosa engega un procés que acaba amb la instal·lació de tots els arxius al nostre sistema:



```

Indicador d'ordres - gem install jekyll bundler
Installing ri documentation for public_suffix-4.0.4
Parsing documentation for addressable-2.7.0
Installing ri documentation for addressable-2.7.0
Parsing documentation for colorator-1.1.0
Installing ri documentation for colorator-1.1.0
Parsing documentation for http_parser.rb-0.6.0
Unknown encoding name "chunked\r\n\r\n25" for ext/ruby_http_parser/vendor/http-parser-java/tools/parse_tests.rb, skipping
Installing ri documentation for http_parser.rb-0.6.0
Parsing documentation for eventmachine-1.2.7-x64-mingw32
Installing ri documentation for eventmachine-1.2.7-x64-mingw32
Parsing documentation for em-websocket-0.5.1
Installing ri documentation for em-websocket-0.5.1
Parsing documentation for concurrent-ruby-1.1.6
Installing ri documentation for concurrent-ruby-1.1.6
Parsing documentation for i18n-1.8.2
Installing ri documentation for i18n-1.8.2
Parsing documentation for ffi-1.12.2-x64-mingw32
Installing ri documentation for ffi-1.12.2-x64-mingw32
Parsing documentation for sassc-2.3.0-x64-mingw32
Installing ri documentation for sassc-2.3.0-x64-mingw32
Parsing documentation for jekyll-sass-converter-2.1.0
Installing ri documentation for jekyll-sass-converter-2.1.0
Parsing documentation for rb-fsevent-0.10.4
Installing ri documentation for rb-fsevent-0.10.4
Parsing documentation for rb-inotify-0.10.1
Installing ri documentation for rb-inotify-0.10.1
Parsing documentation for listen-3.2.1
Installing ri documentation for listen-3.2.1
Parsing documentation for jekyll-watch-2.2.1
Installing ri documentation for jekyll-watch-2.2.1
Parsing documentation for kramdown-2.2.1
Installing ri documentation for kramdown-2.2.1
Parsing documentation for kramdown-parser-gfm-1.1.0
Installing ri documentation for kramdown-parser-gfm-1.1.0
Parsing documentation for liquid-4.0.3
Installing ri documentation for liquid-4.0.3
Parsing documentation for mercenary-0.3.6
Installing ri documentation for mercenary-0.3.6
Parsing documentation for forwardable-extended-2.6.0
Installing ri documentation for forwardable-extended-2.6.0
Parsing documentation for pathutil-0.16.2
Installing ri documentation for pathutil-0.16.2
Parsing documentation for rouge-3.18.0
Installing ri documentation for rouge-3.18.0
Parsing documentation for safe_yaml-1.0.5
Installing ri documentation for safe_yaml-1.0.5
Parsing documentation for unicode-display_width-1.7.0
Installing ri documentation for unicode-display_width-1.7.0
Parsing documentation for terminal-table-1.8.0
Installing ri documentation for terminal-table-1.8.0
Parsing documentation for jekyll-4.0.0
Installing ri documentation for jekyll-4.0.0
Done installing documentation for public_suffix, addressable, colorator, http_parser.rb, eventmachine, em-websocket, concurrent-ruby, i18n, ffi, sassc, jekyll-sass-converter, rb-fsevent, rb-inotify, listen, jekyll-watch, kramdown, kramdown-parser-gfm, liquid, mercenary, forwardable-extended, pathutil, rouge, safe_yaml, unicode-display_width, terminal-table, jekyll after 97 seconds
    
```

Fig. 41: instal·lació de Jekyll

Amb Jekyll ja instal·lat el primer pas és crear l'espai per crear el nostre primer lloc web en local. Per això introduïm a la consola el següent (substituint "myblog" pel nom del nostre blog:

```
jekyll new myblog
```

Això crea la carpeta on tindrem tots els arxius del nostre nou bloc en una carpeta a l'arrel del sistema, és a dir, a C:\myblog en el cas de Windows.

Tot seguit, canviem al nou directori:

```
cd myblog
```

I creem l'estructura bàsica del nostre nou lloc web:

```
bundle exec jekyll serve
```

Si obrim aquesta localització amb l'explorador de Windows podrem veure ja creada l'estructura inicial de tot el lloc web amb els arxius per defecte:

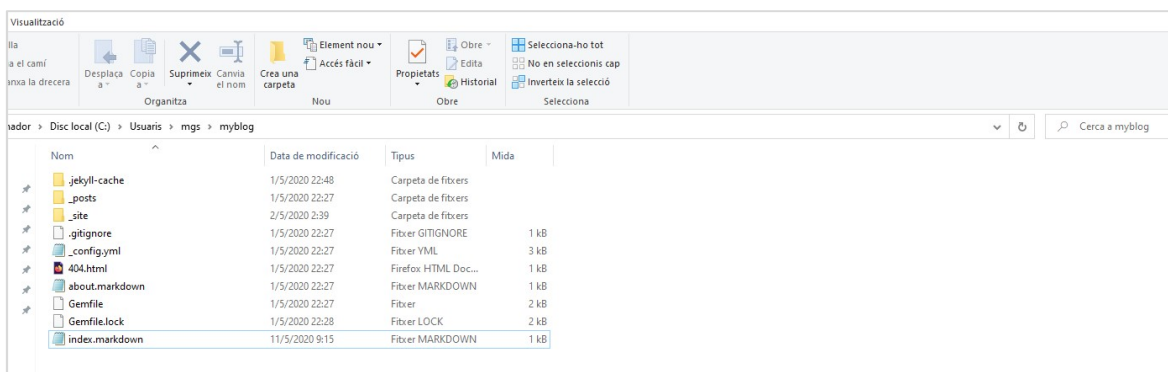


Fig. 42: estructura inicial del nostre blog fet amb Jekyll

A més, amb l'ordre anterior hem posat en marxa el servidor web local integrat a Jekyll que ens permetrà visualitzar directament el nostre nou blog al navegador del nostre ordinador. Ho podem comprovar introduint l'adreça local <http://127.0.0.1:4000/> al nostre navegador:

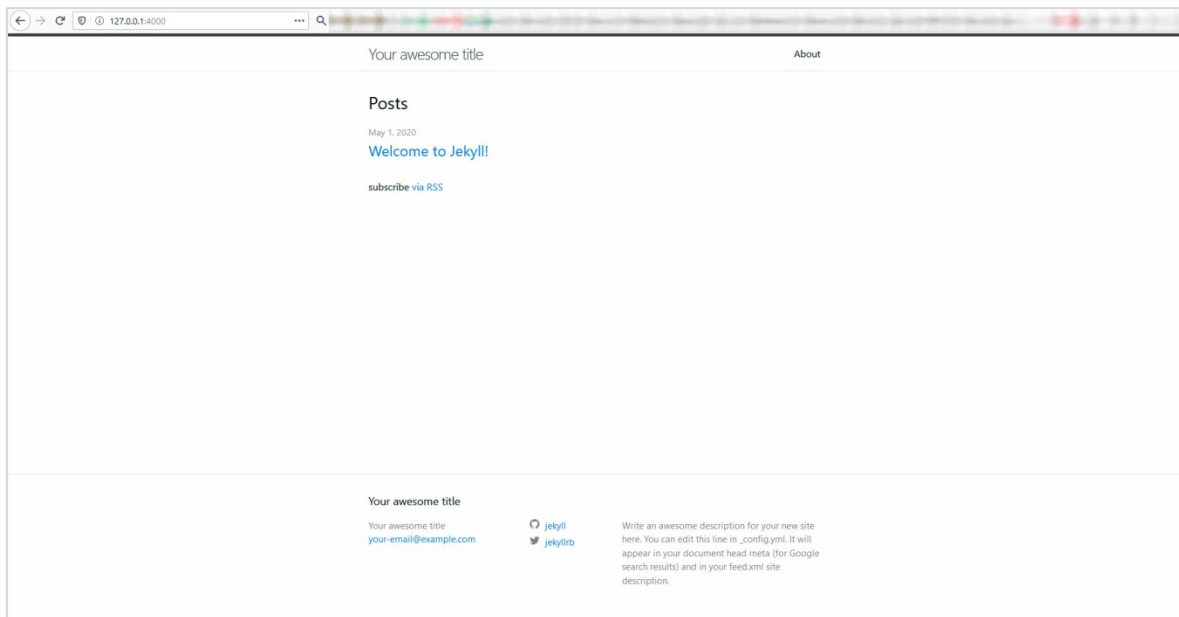


Fig. 43: pàgina per defecte del nostre nou blog amb Jekyll

23.2.4. Markdown

La idea de funcionament de Jekyll és senzilla: escrivim el nostre text amb el format corresponent en un arxiu markdown i automàticament ens apareixerà en el nostre servidor local. com?

Bàsicament hem de localitzar l'arxiu markdown que es correspon amb la pàgina que volem editar. Per exemple pel cas de la pàgina índex és l'arxiu "**index.markdown**" que es troba a l'arrel del nostre blog (recordem C:\myblog en el nostre cas). L'editem amb el nostre editor preferit (la llibreta de Windows serveix), salvem el contingut i recarreguem la pàgina índex local (127.0.0.1) al nostre navegador.

No obstant aquí ens trobem amb el problema més important: com funciona el llenguatge markdow? Tenim dus eines bàsiques per saber-lo

- **Referència de markdown** a internet. Accedim a la pàgina web [markdownguide](#) que on trobarem una àmplia [guia de referència de markdown](#) amb exemples concrets.
- **Editor WYSIWYG markdown** en línia. A la mateixa pàgina web trobem un [editor WYSIWYG markdown](#) en el qual podem posa un codi en markdown i ens el converteix a HTML, de tal manera que ens mostra el propi codi i el resultat final del codi generat. Exemple:

```

Markdown Source:
#Conversió Markdown a HTML
Això és una prova de conversió de Markdown a HTML

Per aquesta prova fem servir el conversor online daringfireball.net que ens permet veure els canvis al moment

...

Per exemple, això és una llista ordenada:

1. Primer element
2. Segon element
3. Tercer element

I això una llista sense ordenar

- Una cosa
- Una cosa totalment diferent

Un altre element possible són els blocs de text

> Per exemple això és un bloc de text realitzat amb Markdown

> Tot el que posem aquí dins serà un bloc diferenciat

...

També podem posar imatges:

!work(https://i0.wp.com/juanjeojeda.com/wp-content/uploads/2017/02/bug-feature-meme.jpg?w=400&ssl=1)

Filter: Markdown Results: Source & Preview Convert
  
```

Fig. 44: codi markdown a l'editor markdownguide

```

HTML Source:
<h1>Conversió Markdown a HTML</h1>
<p>Això és una prova de conversió de Markdown a HTML</p>
<p>Per aquesta prova fem servir el conversor online <a href="https://daringfireball.net">daringfireball.net</a> que ens permet veure els canvis al moment</p>
<hr />
<p>Per exemple, això és una llista ordenada:</p>
<ol>
<li>Primer element</li>
<li>Segon element</li>
<li>Tercer element</li>
</ol>
<p>I això una llista sense ordenar</p>
<ul>
<li>Una cosa</li>
<li>Una cosa totalment diferent</li>
</ul>
<p>Un altre element possible són els blocs de text</p>
<blockquote>
<p>Per exemple això és un bloc de text realitzat amb Markdown</p>
<p>Tot el que posem aquí dins serà un bloc diferenciat</p>
</blockquote>
<hr />
<p>També podem posar imatges:</p>
<p></p>
  
```

Fig. 45: codi HTML generat a partir del codi markdown

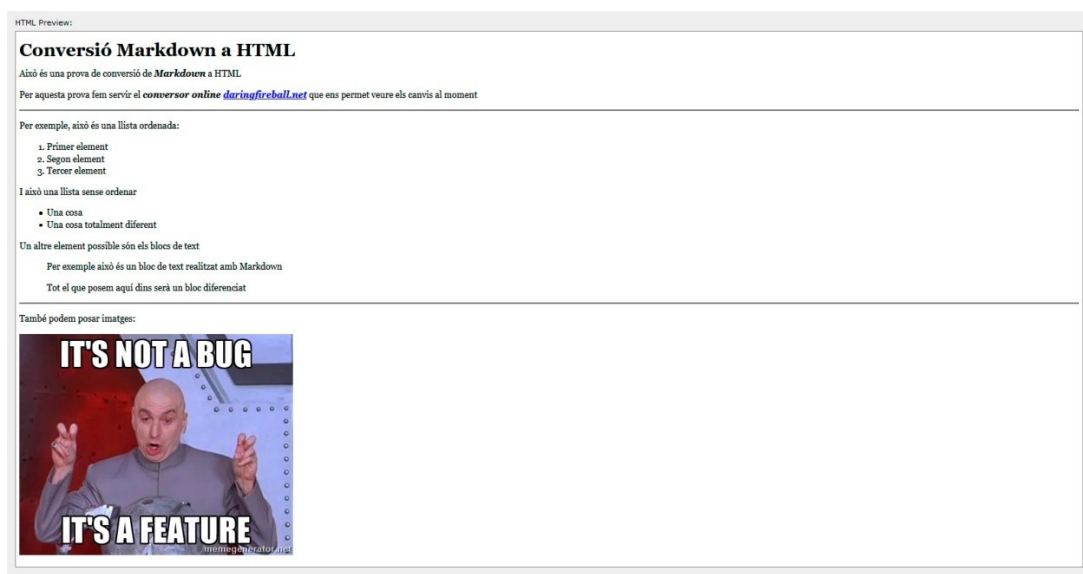


Fig. 46: visualització final del nostre codi markdown al navegador

Les tres pantalles ens apareixen a la mateixa plana, de tal manera que podem anar editant el codi markdown i, clicant un simple botó, ens regenera la previsualització i el codi HTML.

D'aquesta manera l'edició de codi es fa molt més intuïtiva ja que si no no podríem assegurar-nos del resultat dels canvis fins que editéssim l'arxiu a Jekyll , salvéssim els canvis i recarreguéssim el navegador, procés força farragós quan s'ha de fer moltes vegades.

Si posem el codi markdown a l'arxiu "index.markdown" del nostre blog amb Jekyll obtenim el resultat esperat:

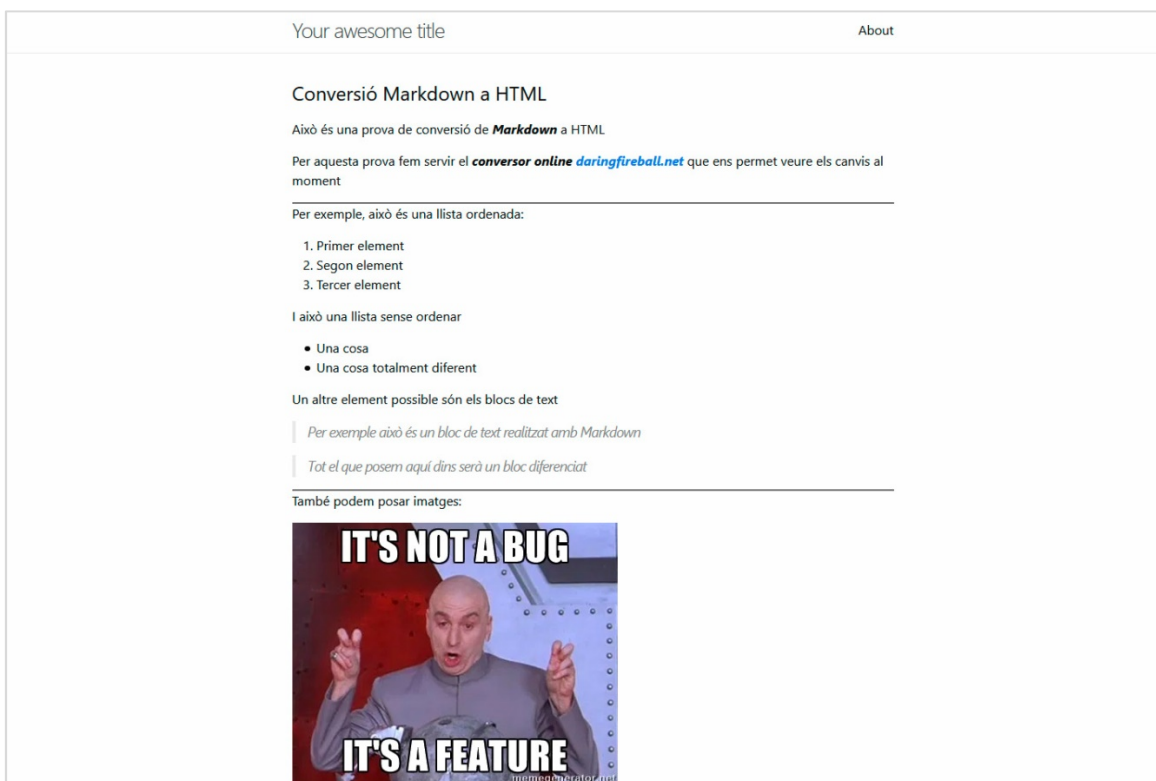


Fig. 47: codi markdown inserta a l'índex del nostre blog amb Jekyll

Cal fer notar que en alguns casos pot variar lleugerament el funcionament de markdown del simulador markdownguide (o fins i tot no funcionar) respecte al que fem servir en la versió final de Jekyll, per la qual cosa és sempre bàsic comprovar el resultat amb el nostre navegador local.

23.2.5. Jekyll vs. Publii

A priori, ens trobem amb dos sistemes iguals que fan el mateix, però la realitat és que es tracta de sistemes totalment diferents que l'únic que tenen en comú és l'objectiu final.

- **Jekyll:** es tracta d'una eina **per a usuaris avançats**. Ja des del moment de la seva instal·lació es veu com calen uns coneixements que van més enllà de l'usuari mitjà, i més enllà fins i tot de la forma de treballar de la majoria de creadors de contingut. L'ús de la línia d'ordres, la obligatorietat de fer servir un llenguatge de marcatge per publicar els continguts, el fet d'haver d'estar familiaritzat amb les estructures dels llocs webs i la generació manual de continguts ens aboquen a una utilització només a l'abast de programadors o usuaris avançats.

L'anàlisi anterior comprèn la simple realització d'un text bàsic per a la pàgina inicial. Si entréssim en la realització de tota l'estructura del lloc web o en la generació de continguts complexos com per exemple galeries de fotografies veuríem que els coneixements necessaris es multipliquen i no estan a l'abast d'aquells que no es dediquen habitualment a la programació.

- **Publii:** tot i ser també un generador de llocs web estàtics ho fa des d'un punt d'aproximació totalment diferent al de Jekyll. Tota l'edició es fa **a través de menús, icones, botons**, etc, i el resultat es veu de forma senzilla mitjançant el generador integrat. De fet, el propi editor de contingut ja dóna una idea molt aproximada del que acabarem veient com a resultat final. La gestió de l'estructura del lloc és totalment visual.

Els enllaços es col·loquen mitjançant llistes desplegable que ja ens mostren els arxius existents, les plantilles es poden instal·lar de manera (relativament) senzilla, etc. És evident que al tractar-se d'una versió beta encara no té el nivell de precisió, estabilitat i facilitat d'ús que té un CMS massiu com pot ser Wordpress, però la idea és poder arribar al mateix nivell i fins i tot superar-lo en quant a senzillesa, cosa que ja ha fet per exemple en temes com la realització de backups del lloc sencer.

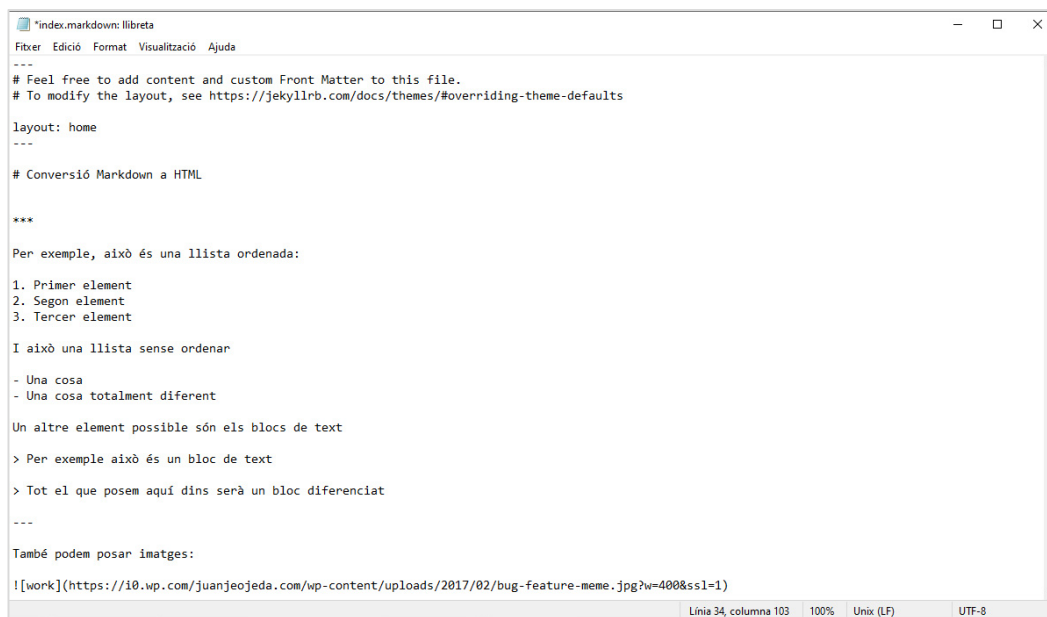


Fig. 48: edició d'un arxiu markdown per a Jekyll (bloc de notes de Windows)

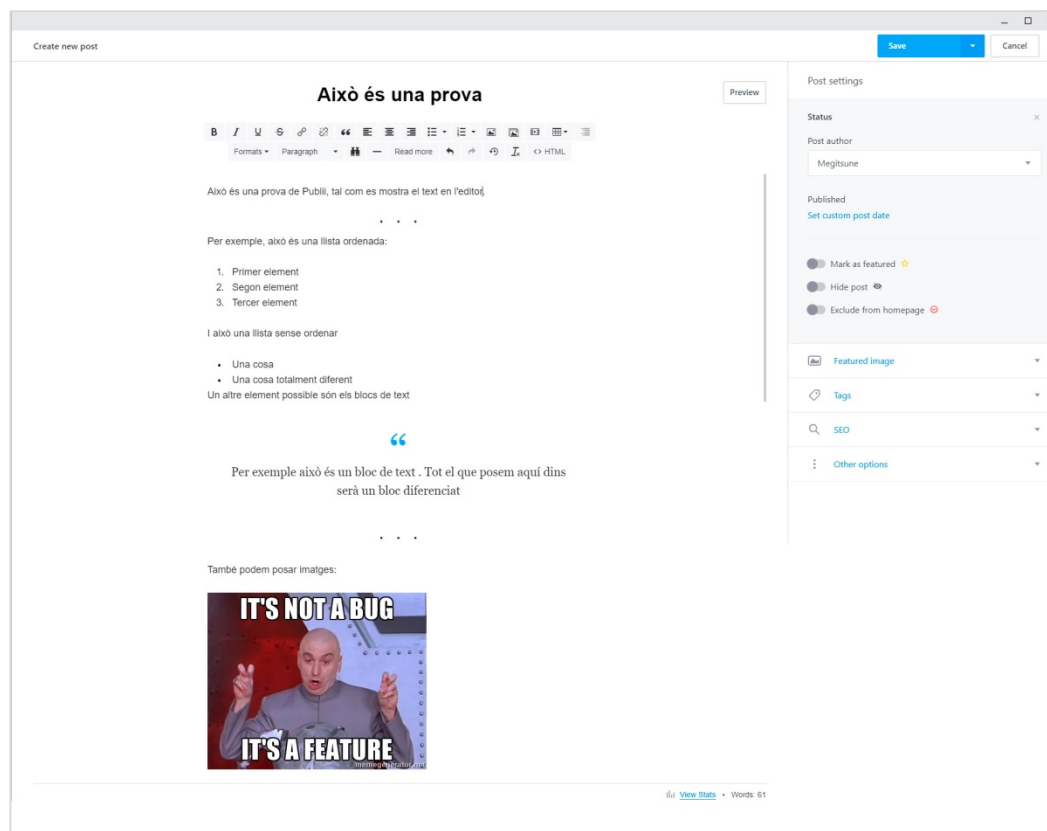


Fig. 49: edició del mateix arxiu anterior fent servir l'editor integrat de Publii

24. Conclusions

Durant tot el treball ja s'ha fet una visió general de què és Publii i quines diferències existeixen amb sistemes semblants. En aquest apartat es resumeixen els resultats obtinguts per fer-los entenedors de manera pràctica.

Per això en primer lloc farem l'exercici de resumir tot aquest treball en unes respostes senzilles i contundents a les principals preguntes realitzades:

Pregunta freqüent	Resposta curta
Un lloc web estàtic ens serveix per les mateixes funcions que un CMS convencional?	No
En cas de voler un lloc web estàtic què és millor, Publii o Jekyll?	Publii
Podem considerar Publii un producte acabat?	No
Val la pena realitzar un lloc web fent servir Publii	Sí
És adequat Publii per a persones sense coneixements de programació (HTML, CSS)?	No

Aquest quadre deixa clars uns conceptes bàsics que han sorgit després de treballar extensivament amb l'eina Publii i comparar-la amb Jekyll (en aquest treball) i amb Wordpress (en altres experiències professionals).

I tot i que la resposta curta a cadascuna d'elles és contundent, la realitat és que hi ha tota una sèrie de matisos que poden arribar a canviar la resposta. Analitzem-los de forma resumida

24.1. Avantatges i desavantatges dels generadors de llocs web estàtics

La generació de llocs web estàtic té dos principals avantatges:

- **Rapidesa del lloc web:** això implica una millor experiència d'usuari, millor posicionament en cercadors i possibilitat de fer-lo funcionar en servidors amb pocs recursos o, el que és el mateix, utilitzar els mateixos recursos per allotjar molts més llocs webs sense caigudes de rendiment, la qual cosa implica abaixar el cost.
- **Senzilla del lloc web:** la senzillesa ens permet centrar-nos més en el contingut que no pas en la tecnologia que fa possible la seva representació. A més fa que els possibles forats de seguretat siguin mínims, la qual cosa implica molt menys

manteniment i simplifica les còpies de seguretat. I una vegada més, reduïm cost de disseny i sobretot de manteniment.

Però evidentment també presenta desavantatges respecte als llocs web dinàmics:

- **Falten sistemes senzills i complets de generació estàtica:** generar un lloc web fent servir arxius "markup", el qual processarem amb un programa que haurem hagut d'instal·lar mitjançant la línia d'ordres del nostre sistema operatiu no sembla una manera intuïtiva i fàcil de fer les coses. Jekyll i similars són molt potents, però el seu públic potencial és reduït. Perquè el sistema de llocs web estàtics es generalitzi en l'ús cal que estigui disponible per als creadors de contingut, no programadors. Publii, un editor WYSIWYG de fàcil instal·lació i gestió és el camí a seguir, però encara té moltes funcions que requereixen endinsar-se en el codi HTML per fer-les funcionar.
- **Els llocs web estàtics no tenen sentit per algunes aplicacions:** com diu el seu nom, tenen sentit per a llocs web que no canvien constantment. Per tant, fer una botiga online amb un generador estàtic no tindria cap mena de sentit (tot i que [hi ha projectes per fer-ho!](#)), com tampoc tindria sentit fer una xarxa social o qualsevol altre servei que es basi en el canvi constant.
- El **procés de renderització** fa més lent el procés de treball, a l'haver de realitzar aquest procés cada vegada que volem veure els resultats al navegador tal com són amb un 100% de fiabilitat, ja que el renderitzat local de Publii no és del tot fiable. No obstant, al final aquesta és la força dels llocs web estàtics, el fet de que la renderització es fa un cop en origen i no els milers o milions de vegades que la pàgina és servida a l'usuari final.

24.2. Procés de creació amb Publii

El procés de creació amb Publii és bo i presenta avantatges evidents que ja hem comentat respecte sistemes que no són **WYSIWYG**, però encara denota falta de maduresa respecte als **CMS convencionals** com Wordpress, Joomla! o Drupal:

- Si bé la creació d'articles senzills on només ens interessa posar text i imatges i un formatat bàsic és satisfactori, en el moment en que vulguem fer una edició que impliqui **capes i disseny CSS** una mica avançat veurem que no tenim cap més remei que recórrer a l'editor de HTML.
- Aquest **editor HTML** és massa senzill, sense pràcticament ajudes a l'igual que l'editor CSS, la qual cosa ens obliga a recórrer a programes externs.
- També ens **falten eines** que ens facin fàcil la gestió d'imatges i arxius ja que la galeria multimèdia pràcticament no té opcions.

En canvi, s'agraeix la gestió integrada de les còpies de seguretat que amb només un parell de clics ens permet fer còpia de tot el lloc web en un arxiu zip o tar que podem agafar i posar on vulguem, fins i tot en un altre ordinador o al núvol per continuar editant des d'un ordinador diferent sense problemes.

A més, cal advertir que just **quan s'estava finalitzant aquest treball, ha aparegut una nova versió de Publlí (0.36)** que incorpora novetats molt importants en el procés de creació. Concretament a partir d'ara existiran tres editors diferents amb els quals podrem treballar:

- **Editor WYSIWYG convencional:** és el que existia fins aquest moment
- **Editor de blocs:** cada paràgraf és un bloc diferenciats, de manera similar com fa Wordpress per defecte en les últimes versions. Cada bloc pot ser tractat de manera totalment independent a la resta

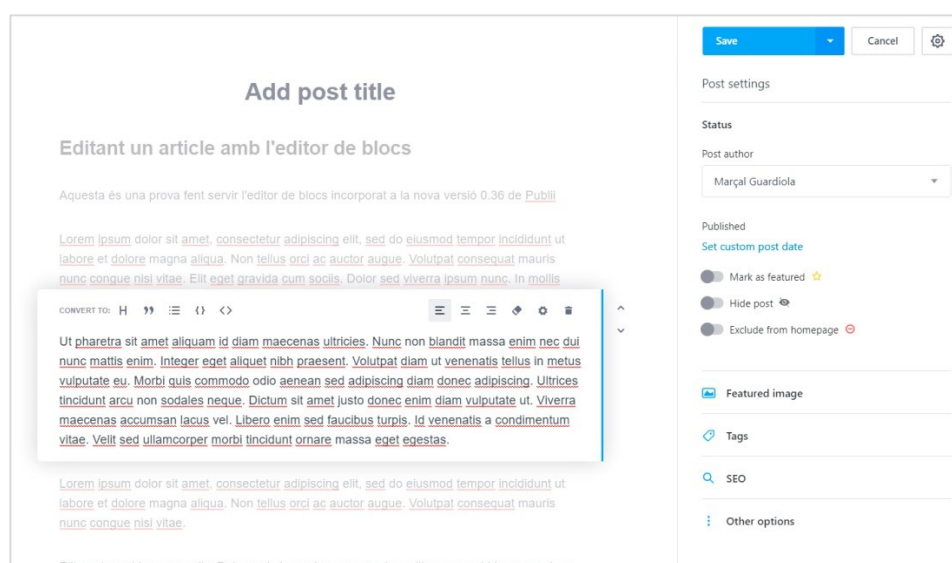
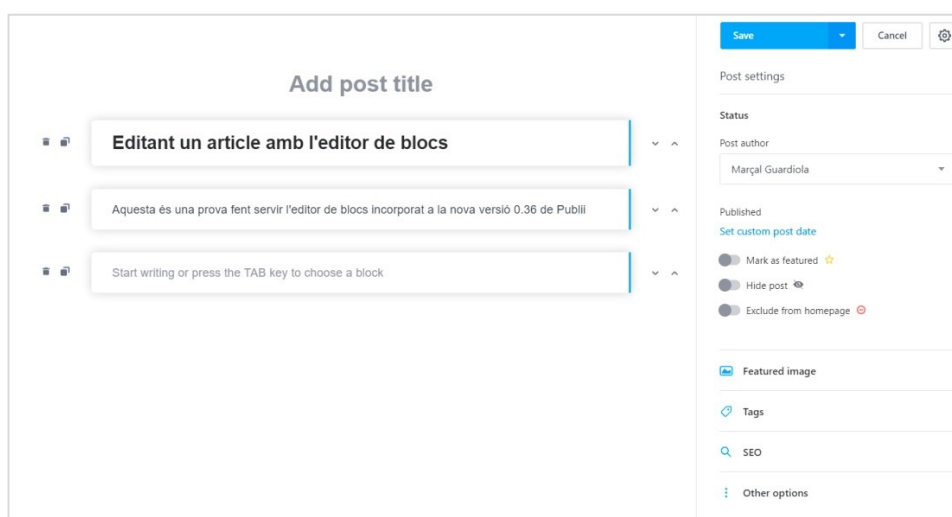


Fig. 50: nou editor de blocs de la versió 0.36 de Publlí

- **Editor markup:** combina la potència gràfica de Publii amb la possibilitat de treballar directament amb markdown (de forma limitada), de manera que aquells desenvolupadors que estiguin acostumats a treballar amb programes com Jekyll ara poden aprofitar el millor d'ambdós mons.

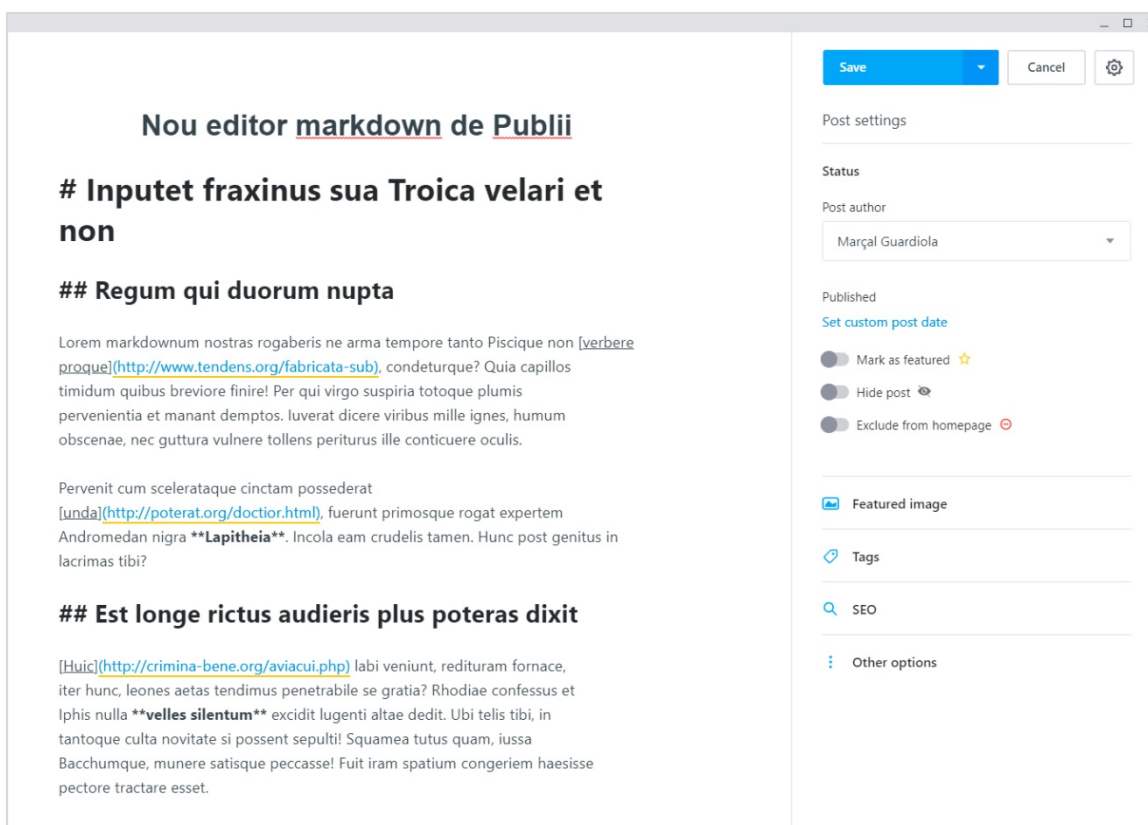


Fig. 51: nou editor markdown integrat a Publii 0.36

24.3. Comparació Publii-Jekyll

Tal com s'ha comentat en l'apartat 24.2, **Publii** ja és ara mateix un generador de llocs web estàtics **perfectament usable** pel generador de continguts mitjà.

És cert que té mancances importants que es mostren quan demanem treball avançat amb capes, objectes, estils CSS, etc, però per una edició bàsica de continguts és perfectament utilitzable per part de l'usuari mitjà. A més, la comunitat que l'està desenvolupant és reduïda però força activa (per exemple durant la realització del treball s'ha alliberat una nova versió del programa), la qual cosa assegura que millorarà amb el temps.

Per la seva banda **Jekyll** (i altres editors de característiques similars) són eines adequades només per a persones que estiguin acostumades a fer servir codi. I la majoria de creadors de contingut no tenen aquest perfil.

De fet possiblement en l'únic camp que podria superar a una eina com Publii és en la rapidesa de crear grans quantitats de continguts de text on el format no sigui molt important, i l'únic que ens interessi sigui la rapidesa d'edició o fins i tot el bolcat automàtic dels mateixos.

Per tant, si **bé la potència de Jekyll és inqüestionable, Publii ens ofereix una interfície visual i una gestió integrada** de plantilles còpies de seguretat, etc, que fa que sigui molt superior a Jekyll per a la major part d'usuaris.

A més, tenint en compte la **incorporació d'un editor markdown dins del propi Publii**, podem dir que Publii porta camí d'integrar tot el que ens pot oferir Jekyll, afegint molts avantatges.

25. Vita Marçal Guardiola Sánchez

Nascut l'any 1974 té el seu primer contacte amb el món de la informàtica a mitjans dels anys 80 gràcies a l'adquisició d'un ordinador de 8 bits **Amstrad CPC 464** que, tot i tenir prestacions molt modestes i limitades pel seu sistema de cinta, permet començar a entendre el món de la informàtica a través del seu llenguatge BASIC integrat.

A mitjans dels anys 90 comença els seus estudis de Disseny Industrial i paral·lelament comença a entrar en el món PC a través d'un 386 amb MS-DOS 5.

L'any 1997 connecta el seu PC a internet i immediatament comença a crear la seva primera pàgina web, que poc després s'anomenaria **WEFER (de Web Ferroviaria)** per tal de conjugar dues de les seves passions: la informàtica (i més concretament la comunicació a través d'internet) i els ferrocarrils. Els que més tard seria WEFER.COM s'allotja a l'espai personal de l'operador REDESTB (desapareguda) i posteriorment a Geocities.



[WEFER.COM](#) es dissenya amb un dels primers editors HTML WYSIWYG que van existir, quan l'habitual encara era dissenyar amb el bloc de notes. Es tractava del [programa AOLPress](#), que funcionava sobre Windows 95.

Mentre realitza els seus estudis de disseny crea diversos llocs webs, el més important dels quals és el [Fòrum del Transport Català](#), que arriba a congrega més de 2.000 usuaris de tot Catalunya i on es parla del món del transport en general

Finalitzada la carrera de disseny industrial comença a treballar en el sector de la enginyeria com a delineant especialitzat en sistemes de telecomunicacions mòbils i fibra òptica, però aviat reorienta el seu camí professional cap al sector de la comunicació.

Combinant diversos llocs de treball i en un món on **internet encara era una gran desconeguda**, es troba amb la situació d'haver de crear llocs web a la vegada que es dedica a realitzar material en paper (revistes, llibres, flyers, etc). Això li dona una visió àmplia i transversal del que és la creació de continguts, fent servir programes tant diferents com **PageMaker i Frontpage, Photoshop o Flash**, havent de gestionar, en el mateix lloc de treball, coneixements tant diversos i allunyats com la instal·lació de servidors web amb la teoria de colors i separació de tintes en la impressió offset.

A partir de 2018 entra en el món del **comerç electrònic**, primer de manera personal a través de sistemes de "partners" i actualment de manera professional, gestionant projectes amb **sistemes Prestashop i Wordpress/Woocommerce**, combinant la creació de continguts amb el posicionament SEO, el màrqueting digital i les xarxes socials.

Annex 1. Lliurables del projecte

Els arxius lliurats a la finalització del projecte són els següents:

- **Memoria_wefer_publii_2020.pdf**: present memòria de tot el projecte en format PDF.
- **wefer.zip**: còpia de tot el lloc web realitzat. Totalment funcional una vegada extret a una carpeta d'un servidor HTML.
- **Presentació en vídeo publicat a l'eina Public@ de la UOC.**
- **Presentació en format lliure**
- **Informe d'autoavaluació**

Annex 2. Codi font (extractes)

Hi ha diverses pàgines de la nova web wefer.com en les quals s'ha hagut de treballar el codi HTML de manera important per donar-li l'aspecte desitjat, diferent del típic llistat vertical per defecte de qualsevol blog

A2.1 - HTML de la pàgina inicial: s'ha dissenyat una estructura que permet posar tres apartats horitzontalment en cada una de les fileres:

```
<p></p>

<div id="contenedor_principal">

<div id="contenedor_apartats"><a href="#INTERNAL_LINK#/post/8"
target="_self"><br>Renfe / ADIF:
<strong>historia</strong></a></div>

<div id="contenedor_apartats"><a href="#INTERNAL_LINK#/post/9"
target="_self"><br>Renfe / ADIF:
<strong>estaciones</strong></a></div>

<div id="contenedor_apartats"><a href="#INTERNAL_LINK#/post/10"
target="_self"><br>Renfe / ADIF: <strong>material
móvil</strong></a></div>

</div><hr>

<p></p>

<div id="contenedor_principal">

<div id="contenedor_apartats"><a href="#INTERNAL_LINK#/post/11"><br>FGC:
<strong>historia</strong></a></div>

<div id="contenedor_apartats"><a href="#INTERNAL_LINK#/post/12"><br>FGC:
<strong>estaciones</strong></a></div>
```

```
<div id="contenedor_apartats"><a href="#INTERNAL_LINK#/post/l3"
target="_self"><br>FGC:
<strong>material mòvil</strong></a></div>
```

```
</div><hr>
```

```
<p></p>
```

En aquest codi podem veure com tota la pàgina s'estructura utilitzant DIVs que es van imbricant uns dins dels altres. A l'antiga pàgina que es substituïa en feia servir l'arcaic sistema de "frames" per tenir la possibilitat de tenir menús fixes i el també antiquat i no recomanat sistema d'utilitzar taules invisibles per fer el posicionament dels elements. el fet de substituir tots aquests elements per DIVs, **en combinació amb el codi CSS**, ens permet controlar millor la posició dels elements i **fer que la pàgina sigui "responsive"** a diferents resolucions, una cosa impossible de controlar només amb les antigues taules.

L'estil dels DIVs es controla mitjançant etiquetes ID mentre que en el cas de les imatges es fa servir l'atribut "class" per definir-les en conjunt.

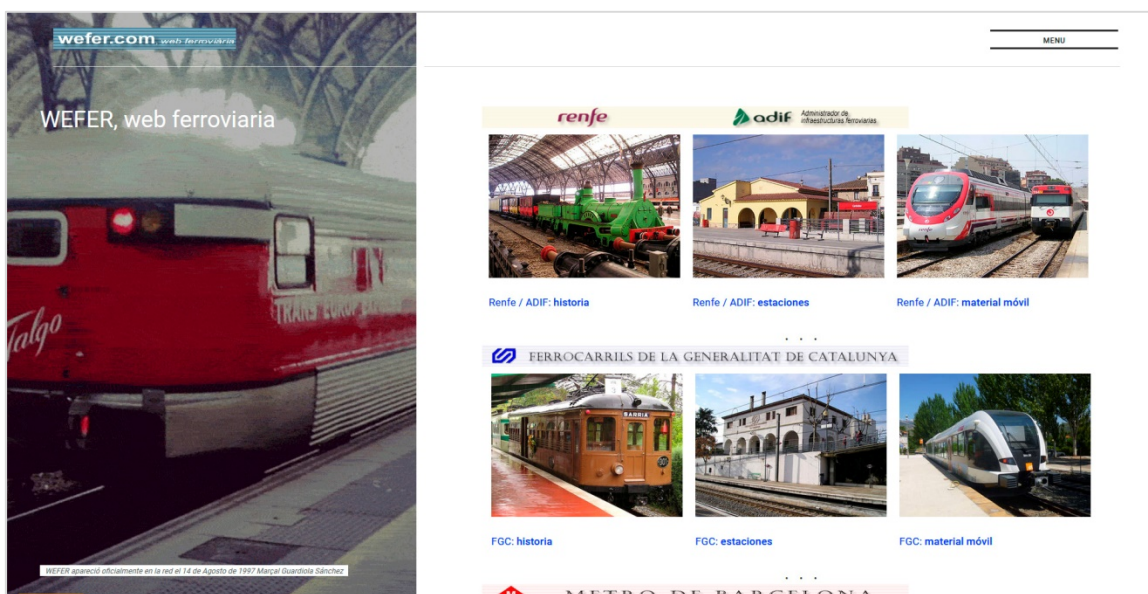


Fig. 52: resultat de la pàgina "index" amb estructura de 3 contenidors horitzontals iguals

A2.2 - HTML de les pàgines d'estacions: Un altre dels codis representatius és el generat per a les pàgines principals de secció on s'enumeren llistat de línies de ferrocarril de manera vertical:

```

<p><em><a href="#"INTERNAL_LINK#/post/17">Inicio</a> --
<strong>FGC: las estaciones</strong></p>
<p> </p>
<p></p>
<hr>
<ul>
<li class="estacio-corr">= estación con correspondencia con otra línea
ferroviaria.</li>
<li class="estacio-norm">= estación sin correspondencia.</li>
</ul>

<div id="contenedor_principal">

<div id="contenedor_2_columnes" class="taula_fgc">
<div id="taula_fgc">
<h4 id="mctoc_1e5ba5ct44">Barcelona Pl. Catalunya &lt;-&gt; Sant
Cugat</h4>
<ul class="estacio-norm">
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/24">Barcelona
Pl. Catalunya (BCN)</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/25">Provença
(BCN)</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/26">Gràcia
(BCN)</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/27">Sant Gervasi
(BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/28">Muntaner (BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/29">La Bonanova (BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/30">Les Tres Torres (BCN)</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/31">Sarrià
(BCN)</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/32">Peu del
Funicular (BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/37">Baixador de Vallvidrera
(BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/33">Les Planes (BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/34">La Floresta</a></li>
<li><a href="#"INTERNAL_LINK#/post/35">Valldoreix</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/36">Sant
Cugat</a></li>
</ul>
</div>
</div>

<div id="contenedor_2_columnes" class="taula_fgc">
<div id="taula_fgc">
<h4 id="mctoc_1e5ba4b2d1">Gràcia &lt;-&gt; Av. Tibidabo</h4>
<ul class="estacio-norm">
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/26">Gràcia
(BCN)</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/38">Plaça Molina
(BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/39">Pàdua (BCN)</a></li>
<li><a href="#"INTERNAL_LINK#/post/40">El Putxet (BCN)</a></li>
<li class="estacio-corr"><a href="#"INTERNAL_LINK#/post/41">Avinguda
Tibidabo (BCN)</a></li>
</ul>
</div>

```

```

<br>
<div id="taula_fgc">
<h4 id="mctoc_le5ba4bhq2">Sarrià &lt;-&gt; Reina Elisenda</h4>
<ul class="estacio-norm">
<li class="estacio-corr"><a href="#INTERNAL_LINK#/post/26">Sarrià (BCN)</a></li>
<li><a href="#INTERNAL_LINK#/post/42">Reina Elisenda (BCN)</a></li>
</ul>
</div>
</div>
</div>

```

En aquest codi podem veure com els panells on s'indiquen els llistat d'estacions que formen les línies han estat definits també amb **DIVs imbricats**, que ens permeten fer servir aquesta pàgina amb un sistema de presentació de dues columnes per aprofitar millor l'espai.

També és interessant observar el codi generat per Publii per tal d'enllaçar els articles interns del lloc web, identificats com **"#INTERNAL_LINK#"** i que després, en la fase de renderització, seran substituïts pels enllaços adequats del lloc web. És important tenir en compte aquest aspecte perquè si, pel que fos, agaféssim codi del nostre lloc web ja publicat i el copiéssim a Publii, aquest codi no funcionaria correctament ja que Publii no els reconeixeria com a enllaços interns i en cas de canvi de nom quedarien trencats.

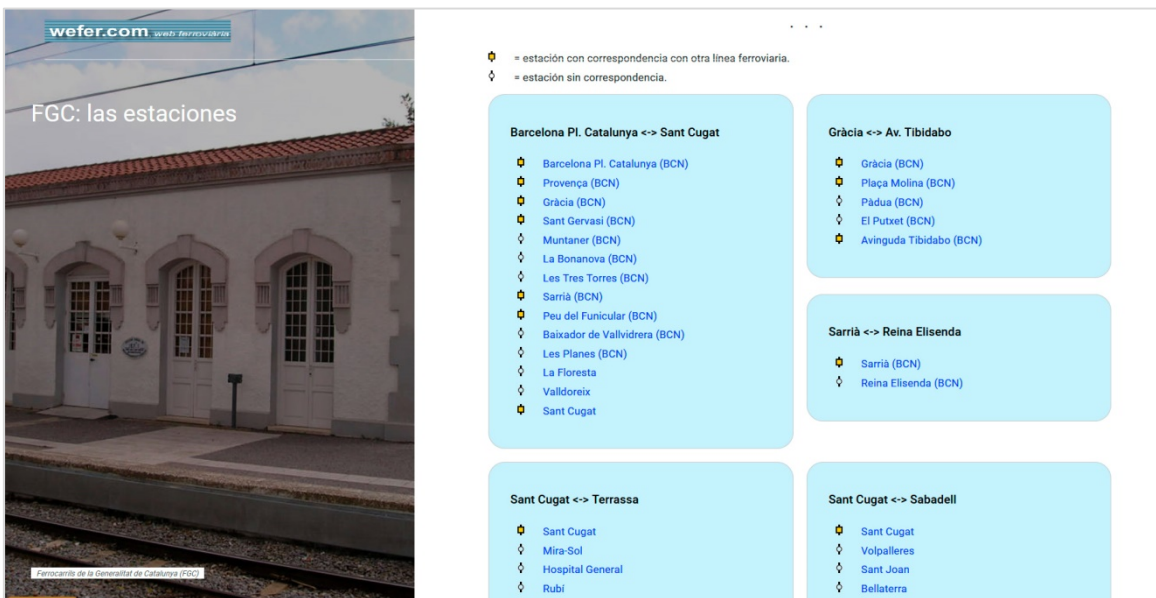


Fig. 53: pàgina "Estacions FGC" amb llistat de dos contenidors i llistes personalitzades

A2.3 – HTML de les pàgines d'unitats: en aquestes pàgines es repeteix un patró de quatre seccions en horitzontal, amb la tercera que conté el text principal i per tant necessita ser més gran que tota la resta:

```

<p><em><a href="#INTERNAL_LINK#/post/17">Inicio</a> --&gt;
</em><strong>Renfe / ADIF: el material mòbil</strong></p>
<hr>
<h2 id="mctoc_1e7e9a0di0" class="align-center">Locomotoras eléctricas
de Renfe </h2>
<p> </p>
<hr class="separador_01">
<div class="contenedor_principal">
<div class="contenedor_4_columnes_t1"> </div>
<div class="contenedor_4_columnes_t2">Nº Serie Renfe</div>
<div class="contenedor_4_columnes_t3"> </div>
<div class="contenedor_4_columnes_t4"> </div>
<hr class="separador_01">
<div class="contenedor_principal">
<div class="contenedor_4_columnes_t1"><a
href="#INTERNAL_LINK#/post/106"></a></div>
<div class="contenedor_4_columnes_t2">1 a 7</div>
<div class="contenedor_4_columnes_t3">
<p class="estil_01"><strong>Años de servicio: </strong>1907 a 1966.
<strong>Unidades: </strong>8.</p>
Unas de las primeras locomotoras eléctricas de España. Construidas para
un duro tramo de la línea <strong>Linares-
Almería.</strong> Eran <strong>trifásicas</strong>, con catenaria de
dos cables.</div>
<div class="contenedor_4_columnes_t4"><a
href="#INTERNAL_LINK#/post/106"><button class="boto_01">Leer más
&gt;</button></a></div>
</div>
<hr class="separador_01">
<div class="contenedor_principal">
<div class="contenedor_4_columnes_t1"></div>
<div class="contenedor_4_columnes_t2">21 a 24</div>
<div class="contenedor_4_columnes_t3">
<p class="estil_01"><strong>Años de servicio: </strong>1963 a
1966. <strong>Unidades: </strong>4.</p>
Locomotoras adquiridas para dar servicio de viajeros a la línea
Almería-Nacimiento por donde ya circulaba la <strong>serie 1 a
7</strong>. Fueron locomotoras trifásicas con catenaria de dos
hilos.</div>
<div class="contenedor_4_columnes_t4"><button class="boto_01">Leer más
&gt;</button></div>
</div>
<hr class="separador_01">

```

En aquest codi veiem com existeix una estructura de contenidor principal <div class="contenedor_principal"> que conté quatre divs més, cadascun amb una classe diferent que respon a les definicions fetes en l'arxiu d'estils generals CSS. Aquesta estructura es va repetint tantes vegades com fitxes existeixen.

També cal esmentar com alguns textos són modificats també mitjançant l'aplicació d'estils com per exemple <p class="estil-01">.

El resultat és el següent:

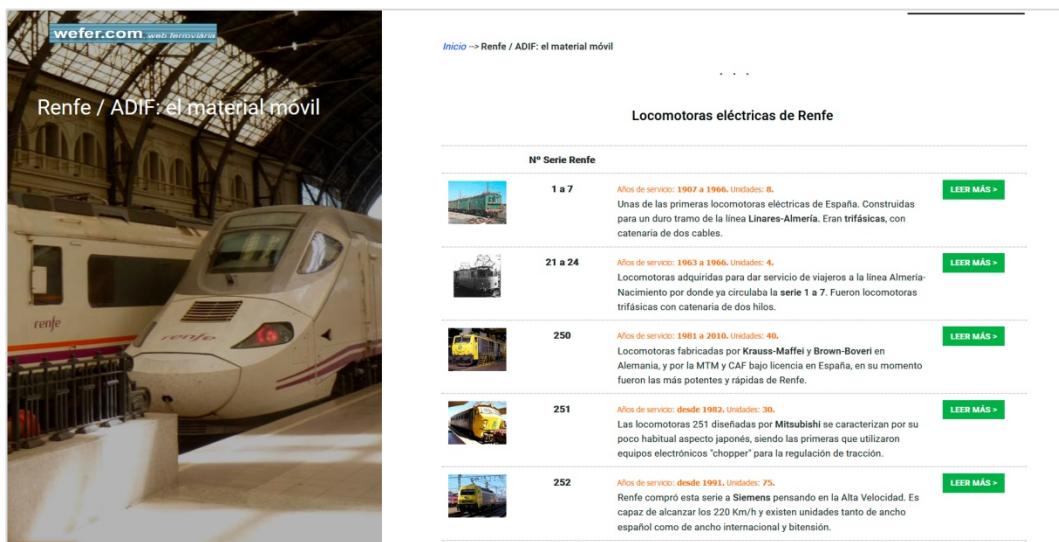


Fig. 54: pàgina "Unitats Renfe" amb llistat horitzontal de 4 contenidors de diferents mides



Fig. 55: la mateixa pàgina de la fig. 51, en versió mòbil

A2.4 – Estils CSS: pel que fa al codi **CSS**, aquest s'ha generat en bona part amb **Dreamweaver** pel fet que aquest programa disposa d'un editor amb múltiples ajudes no disponibles al bàsic editor de Publlii, Després s'ha enganxat i acabat de polir en l'editor CSS de Publlii i recull característiques que modifiquen els CSS per defecte de la plantilla *Square*:

```
@charset "utf-8";
/* CSS Document */
/* Marçal Guardiola Sánchez. Barcelona, 2020 */
/* ----- Estils generals ----- */

A:hover {color: #FF0000; text-decoration: underline}
    .estil-01 { color: #F60; padding: 0%; margin-top: 0.2em; margin-bottom: 0.2em; font-
family:Tahoma, Geneva, sans-serif; font-size:0.8em; font-weight:bold;}

    body { background-image: url(/media/files/reixa-negr.gif); }
    .capsalera { font-weight:bolder; }
    .estacio-corr { list-style-image: url('/media/files/estacio-correspondencia.gif'); }
    .estacio-norm { list-style-image: url('/media/files/estacio-normal.gif'); }
    table {
        white-space: normal; /*Elimina la barra de scroll (ve de definició de plantilla) i adapta
el text de la taula */
    }
    .top {
border-bottom: none; /*Eliminem ratlla blanca sota logo*/
    }

/* ===== ESTILS ALTA RESOLUCIO ===== */

@media all and (min-width: 56.25em) {
    .main .tank > div, .main__right > div {
        max-width: 90rem;
    }
    .main .tank, .main__right {
width: 63%;
    }
    .main__left {
width: 35vw;
    }
    .top {
        width: 60% !important;
    }
    .menu-toggle {
        height: 35px;
width: 12rem;
text-indent: 0rem;
border-top: none;
padding: 1%;
    }
}
```

```

/*----- Estructura -----*/

    .contenedor_principal { overflow:hidden; }
    .contenedor_2_columnes, .contenedor_3_columnes, .contenedor_4_columnes,
.contenedor_4_columnes_t1, .contenedor_4_columnes_t2, .contenedor_4_columnes_t3,
.contenedor_4_columnes_t4 {
    display:inline-table;
    padding:1%;
    overflow:hidden;
}

    .contenedor_2_columnes { width: 49% }
    .contenedor_3_columnes { width: 32% }
    .contenedor_4_columnes { width: 24% }
.contenedor_4_columnes_t1, .contenedor_4_columnes_t2, .contenedor_4_columnes_t3,
.contenedor_4_columnes_t4 {
    text-align: center;
}

    .contenedor_4_columnes_t1 { width: 12%;}
    .post__image { padding: 0%; margin-top: 0%; }

.contenedor_4_columnes_t2 { width: 17%; font-size: 1em; font-weight: bold;}
.contenedor_4_columnes_t3 { width: 55%; text-align:left;}
.contenedor_4_columnes_t4 { width: 14%; }

    .boto_01 {
        background-color: #4CAF50;
        border: none;
        color: white;
        padding: 0.4rem;
padding-left: 0.8rem;
padding-right: 0.8rem;
margin 0px;
        text-align: center;
        text-decoration: none;
        font-size: 0.8rem;
font-weight: bold;
        cursor: pointer;
    }

    .boto_01:hover {
        color: #ffffff;
background-color:#F00;
    }

    .separador_01 {
        border: 1px dotted grey;
margin: 0rem;
}

    .separador_01::before { /* Desactivacio del parametre ::before per a hr del llistat*/
display: none;
}

```



```

.taula_fg {
    background: #D1F1FC;
    border:1px solid #ccc;
    border-radius: 1.5rem;
    padding-bottom: 2.5rem;
    text-align: left;
    padding-left: 2rem;
}

}

/* ===== ESTILS BAIXA RESOLUCIO ===== */

@media all and (max-width: 56.1875em) {
    .menu-toggle {
        height: 1.5rem;
        width: 6rem;
        text-indent: 0rem;
        border-top: none;
        padding: 1%;
        background: #ffffff;
    }
    .contenedor_principal { overflow:hidden; }

    .contenedor_2_columnes, .contenedor_3_columnes, .contenedor_4_columnes,
    .contenedor_4_columnes_t1, .contenedor_4_columnes_t2, .contenedor_4_columnes_t3,
    .contenedor_4_columnes_t4 {
        display:inline-table;
        padding:1%;
    }

    .contenedor_2_columnes { width: 49%; }
    .contenedor_3_columnes { width: 32%; }
    .contenedor_4_columnes { width: 24%; }
    .contenedor_4_columnes_t1 { width: 20%; }
    .post__image { padding: 0%; margin-top: 0%; }
    .contenedor_4_columnes_t2 { width: 18%; font-size:large; font-weight:bold; }
    .contenedor_4_columnes_t3 { width: 60%; text-align:left; }
    .contenedor_4_columnes_t4 { width: 0%; display:none;}

    .taula_fg {
        background: #D1F1FC;
        border:1px solid #ccc;
        border-radius: 1rem;
        padding-bottom: 1rem;
        padding-top: -2rem;
        text-align: left;
        padding-left: 0.2rem;
        padding-right: 0.2rem;
    }
    li {
        position: relative;
        text-indent: -1rem;
        margin-left: -0.4rem;
    }
}
}

```

Veiem diverses definicions que es refereixen tant a l'estil i proporcions dels DIVs com a l'estil de llistes i similars. Hem de recordar que, de moment en la versió actual de Publii aquests estils **poden presentar petites diferències** entre el que apareix en la renderització en local del lloc web amb la que es presenta finalment en el lloc web publicat.

A més podem veure que hi ha tres grans seccions:

- **Estils generals:** defineix els estils comuns a tots els dispositius
- **Estils alta resolució:** defineix els estils per a dispositius amb resolucions grans, habitualment ordinadors de sobretaula i portàtils
- **Estils baixa resolució:** defineix els estils per a dispositius de baixa resolució, de pantalla petita, normalment tauletes i mòbils. Es redefeixen els marges, les dimensions dels DIVs de posicionament i fins i tot s'amaguen elements per fer més agradable la visualització en dispositius de pantalla petita.

Annex 3. Llibreries/Codi extern utilitzat

Per a la realització d'aquest projecte no s'han utilitzat llibreries externes més enllà del programari referit més detalladament en l'apartat 8, és a dir:

- Sistema operatiu **Windows 10** com a mètode principal de desenvolupament
- Sistema operatiu **Linux Mint** com a plataforma per provar la versió Linux del programa Publii.
- **Publii**: base del treball actual, ens permet generar el lloc web estàtic en local i quan el tenim acabat pujar-lo al web.
- **Adobe Photoshop CS6**: per al tractament d'imatges, generació de GIFs animats i esquemes.
- **Adobe Dreamweaver CS6**: en alguns casos com l'edició CSS o la creació de DIVs les eines WYSIWYG de Publii són insuficients i/o generen resultats no desitjats. en aquests casos la solució ha estat ajudar-se de Dreamweaver per tal de crear el codi i posteriorment enganxar-lo a Publii en format HTML.
- **Photomarks**: programa d'edició d'imatges en lot. Permet realitzar tasques complexes com l'adició de marques d'aigua, amb controls de posició avançats, de manera ràpida a centenars o milers d'imatges a la vegada.
- **FileZilla**: client FTP utilitzat puntualment per tal de fer neteja d'arxius que ja no s'utilitzen.

Annex 4. Captures de pantalla

En aquest annex es recopilen les captures de pantalla relatives a diferents parts del procés de creació del lloc web:

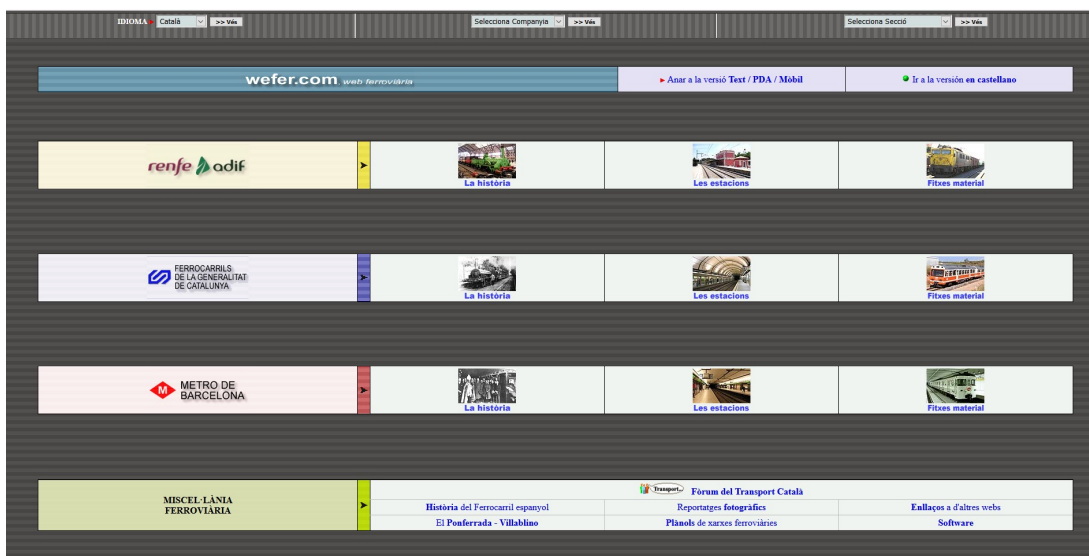


Fig. 01: aspecte actual de la pàgina inicial, realitzada amb taules i "frames"

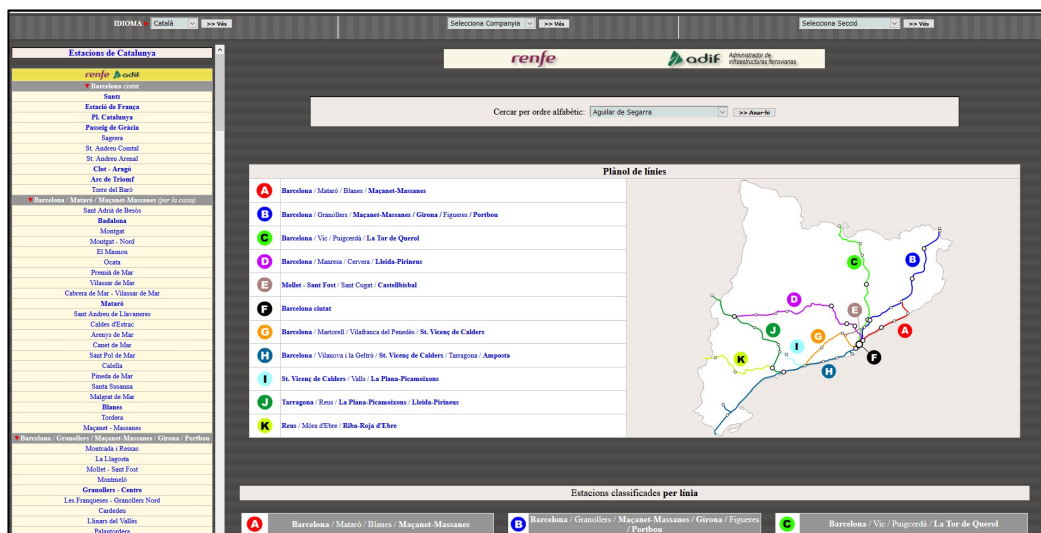


Fig. 02: aspecte actual de la pàgina d'estacions a remodelar

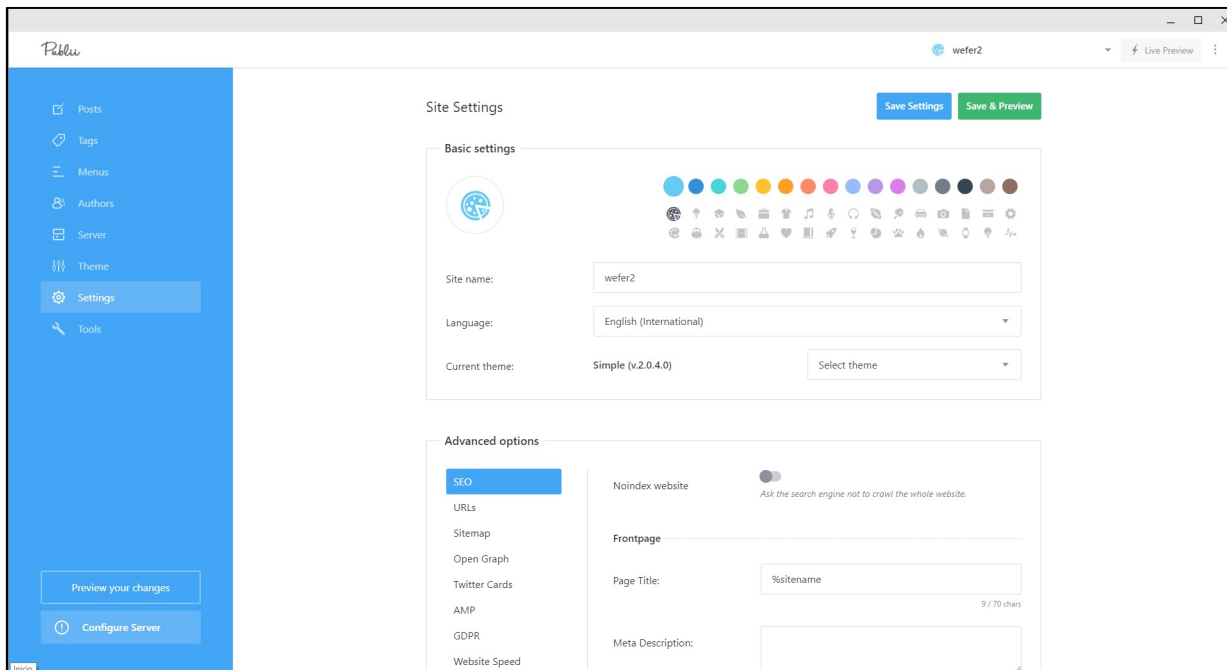


Fig. 03: interfície de l'aplicació Publii a Windows 10

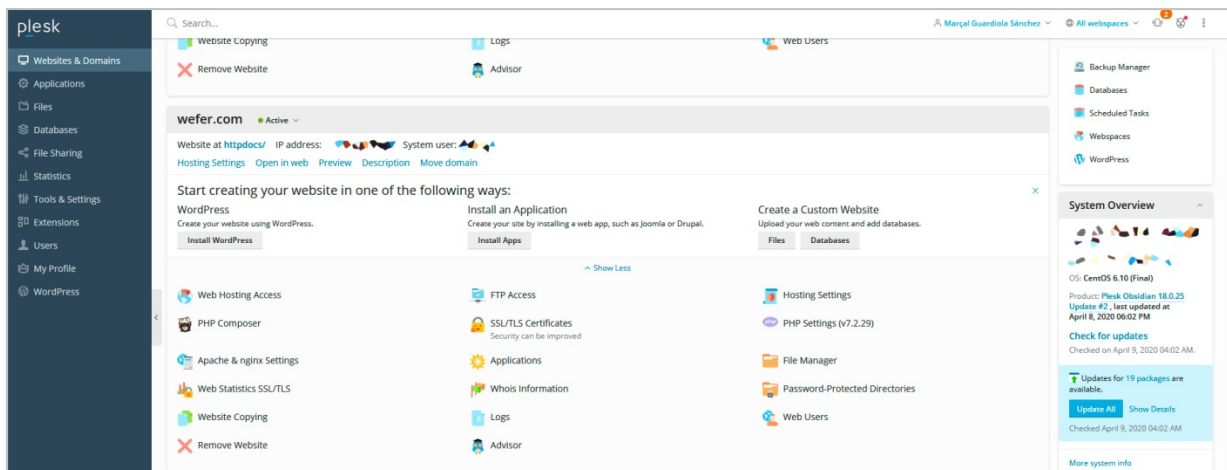


Fig 07: PLESK, un dels gestors que ens permet controlar els paràmetres del nostre lloc web

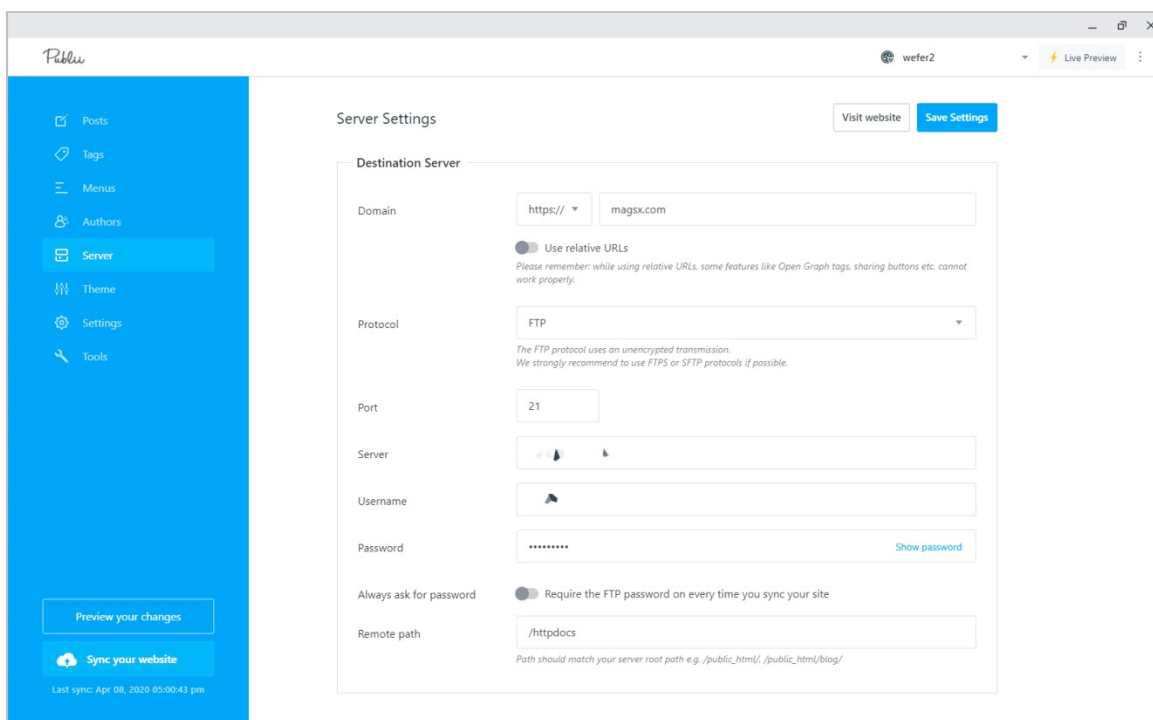


Fig 11: aspecte e la finestra de configuració del servidor on publicarem el nostre web

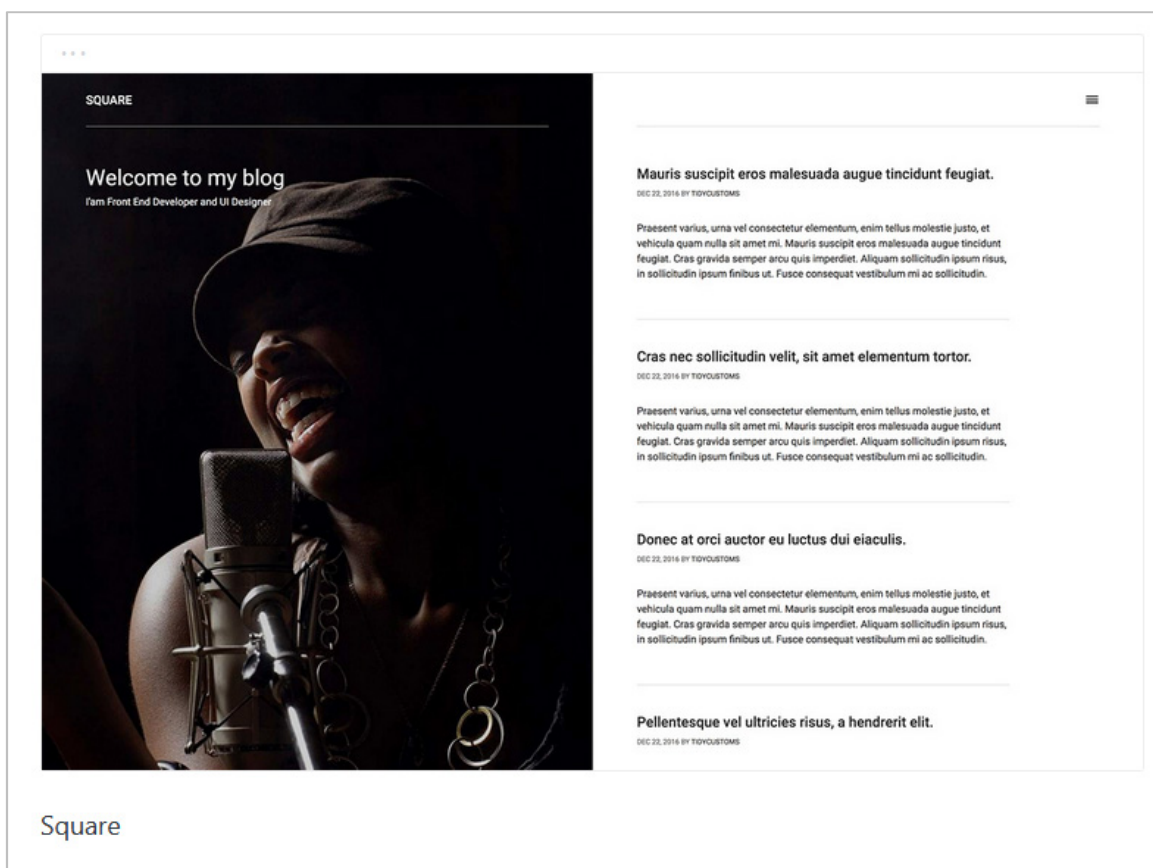


Fig 13: plantilla SQUARE per a Publii

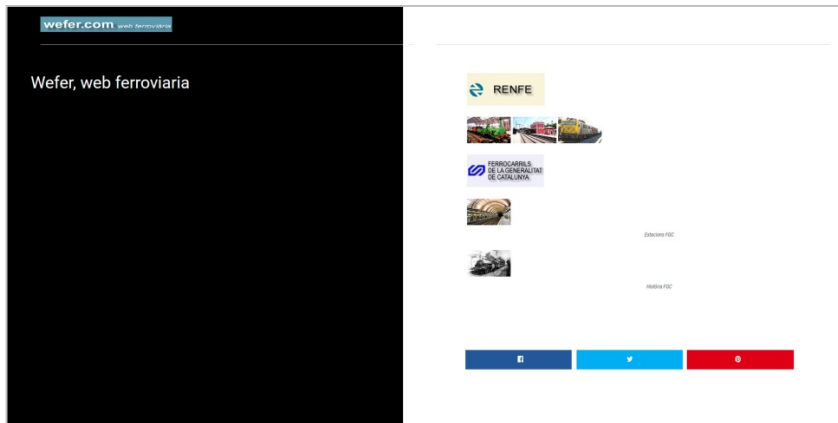


Fig 15: començament de l'adaptació de la plantilla Square

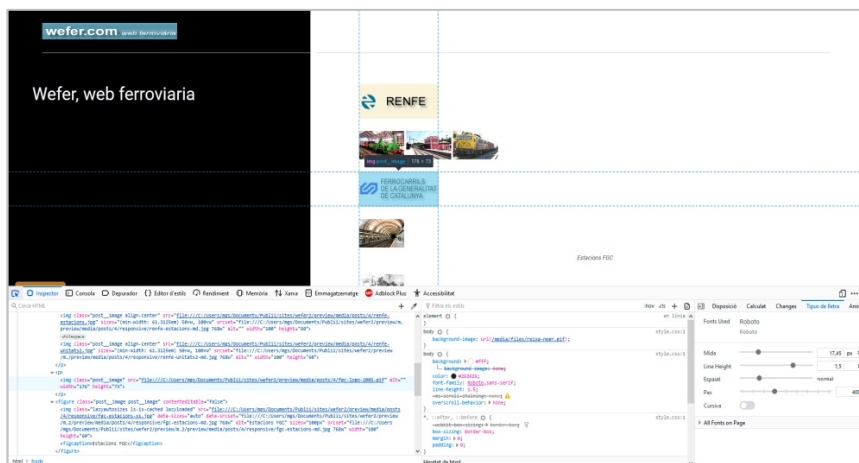


Fig 16: exploració del codi de la pàgina utilitzant "Firefox Developer Tools"

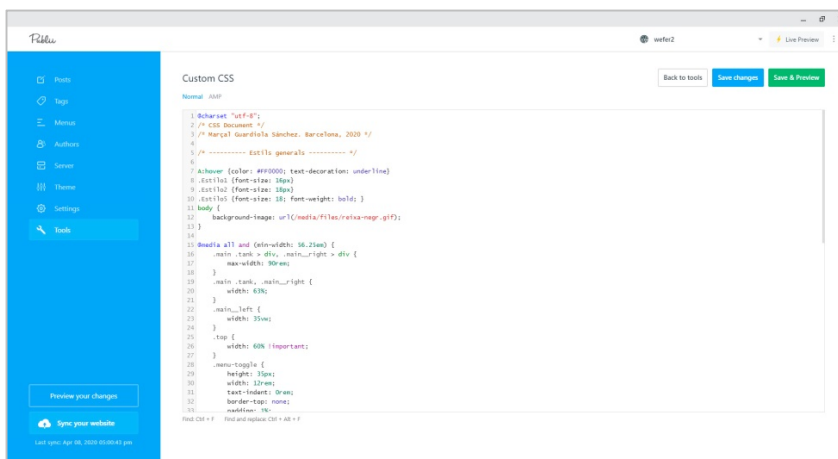


Fig 17: Codi CSS personalitzat

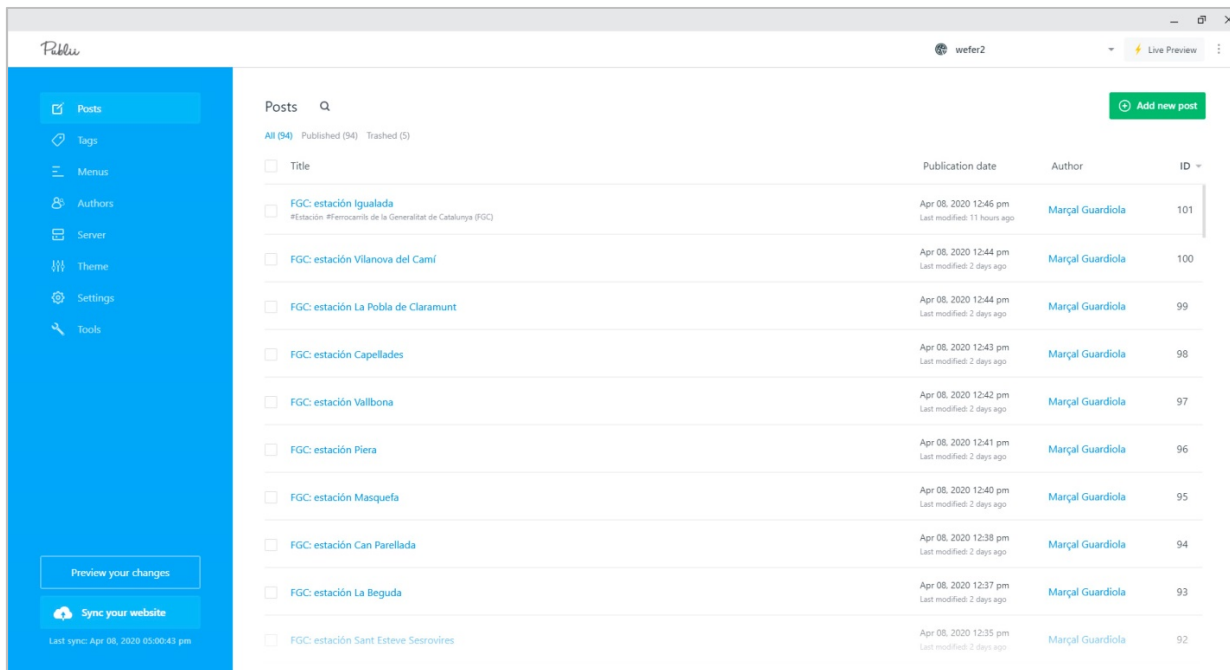


Fig 18: Llistat d'articles

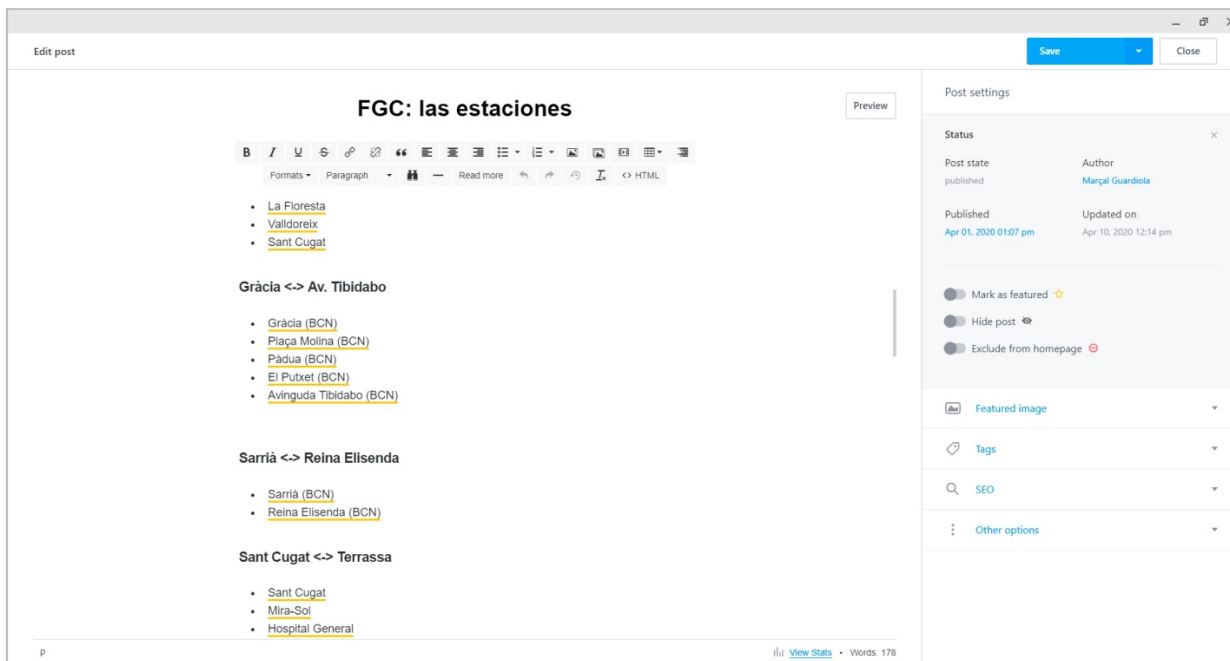


Fig 19: finestra d'edició d'articles en mode WYSIWYG


```

Edit post
Apply changes Cancel
10 <div id="contenedor_2_columes" class="taula_fgc">
11 <div id="taula_fgc">
12 <h4 id="mctoc_Le56a5ct44">Barcelona P1, Catalunya &lt; &gt; Sant Cugat </h4>
13 <ul class="estacio-norm">
14 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/24">Barcelona P1, Catalunya (BCN) </a></li>
15 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/25">Provència (BCN) </a></li>
16 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/26">Gràcia (BCN) </a></li>
17 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/27">Sant Gervasi (BCN) </a></li>
18 <li><a href="#INTERNAL_LINK#post/28">Sant Joan de Vilatorrada (BCN) </a></li>
19 <li><a href="#INTERNAL_LINK#post/29">La Bonanova (BCN) </a></li>
20 <li><a href="#INTERNAL_LINK#post/30">Les Tres Torres (BCN) </a></li>
21 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/31">Sarrià (BCN) </a></li>
22 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/32">Peu del Funicular (BCN) </a></li>
23 <li><a href="#INTERNAL_LINK#post/33">Baixador de Vallvidrera (BCN) </a></li>
24 <li><a href="#INTERNAL_LINK#post/34">Les Planes (BCN) </a></li>
25 <li><a href="#INTERNAL_LINK#post/35">La Floresta (BCN) </a></li>
26 <li><a href="#INTERNAL_LINK#post/36">Sant Cugat (BCN) </a></li>
27 </ul>
28 </div>
29 </div>
30 </div>
31 <div id="contenedor_2_columes" class="taula_fgc">
32 <div id="taula_fgc">
33 <h4 id="mctoc_Le56a462d1">Gràcia &lt; &gt; Av. Tibidabo </h4>
34 <ul class="estacio-norm">
35 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/28">Gràcia (BCN) </a></li>
36 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/38">Plaça Molina (BCN) </a></li>
37 <li><a href="#INTERNAL_LINK#post/39">Pàdua (BCN) </a></li>
38 <li><a href="#INTERNAL_LINK#post/40">El Putxet (BCN) </a></li>
39 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/41">Avinguda Tibidabo (BCN) </a></li>
40 </ul>
41 </div>
42 </div>
43 <div id="taula_fgc">
44 <h4 id="mctoc_Le56a4bq2">Sarrià &lt; &gt; Reina Elisenda </h4>
45 <ul class="estacio-norm">
46 <li class="estacio-corr"><a href="#INTERNAL_LINK#post/26">Sarrià (BCN) </a></li>
47 <li><a href="#INTERNAL_LINK#post/42">Reina Elisenda (BCN) </a></li>
48 </ul>
49 </div>
50 </div>
51 </div>
52 <div id="contenedor_principal">
53 <div id="contenedor_2_columes" class="taula_fgc">
54 <div id="taula_fgc">

```

Fig 20: finestra d'edició d'articles en mode codi HTML

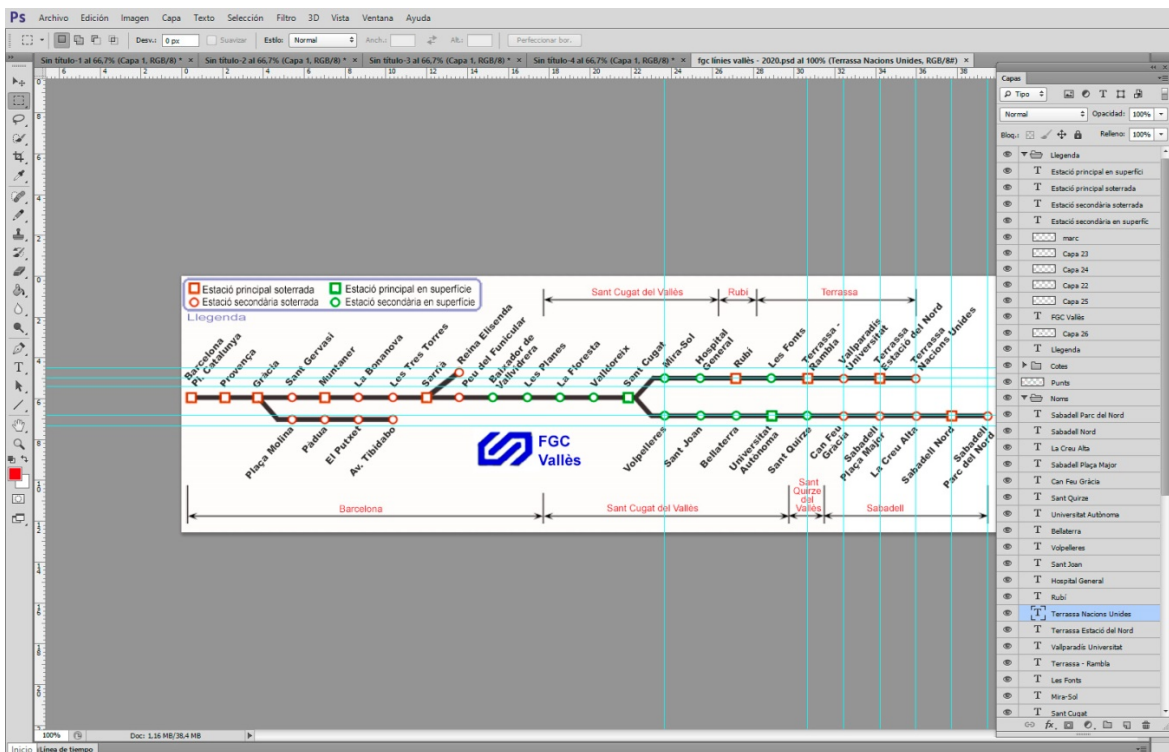


fig 22: procés de treball amb Photoshop

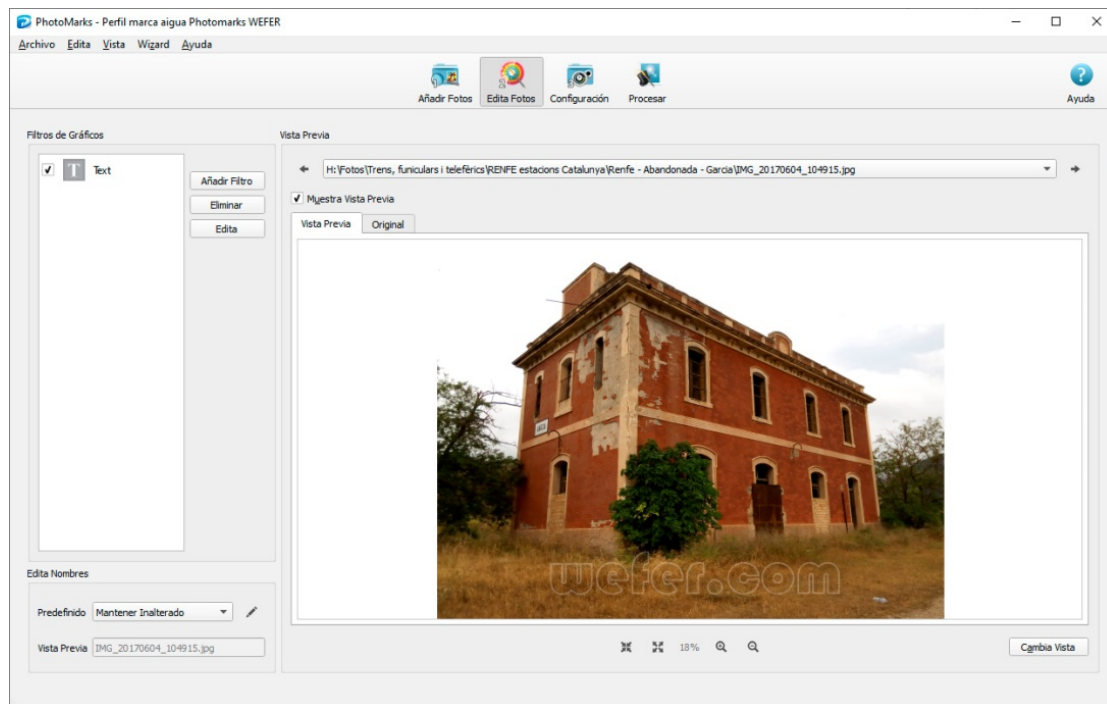
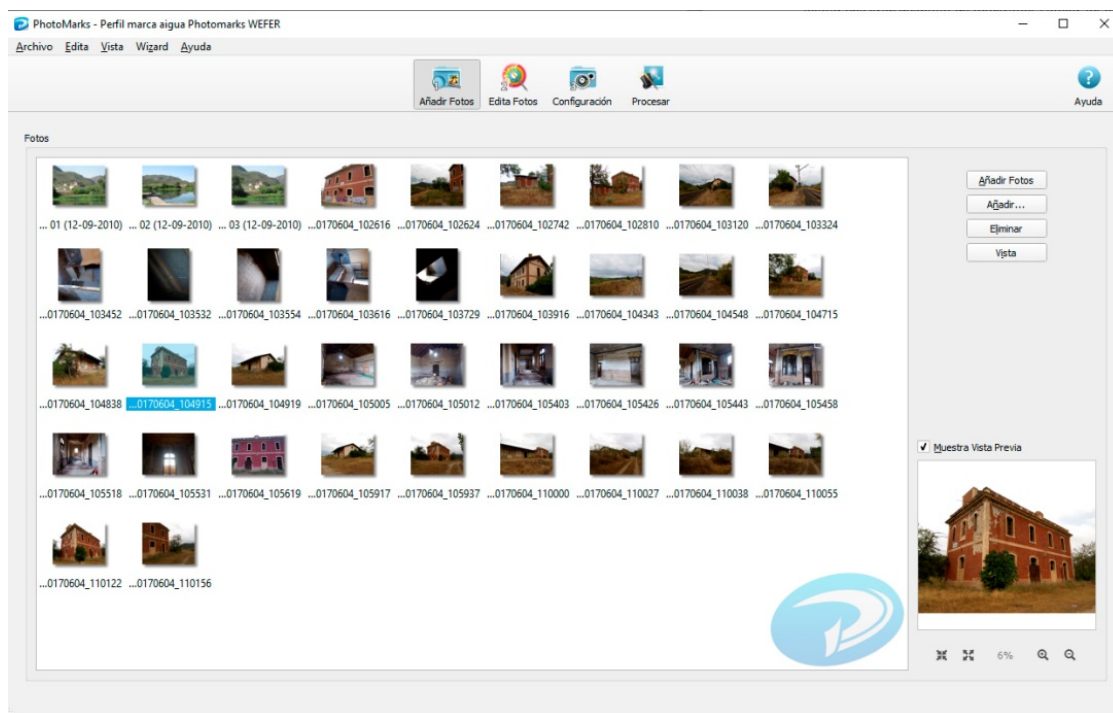


Fig 23: Inclusió d'una marca d'aigua en un conjunt d'imatges amb PhotoMarks

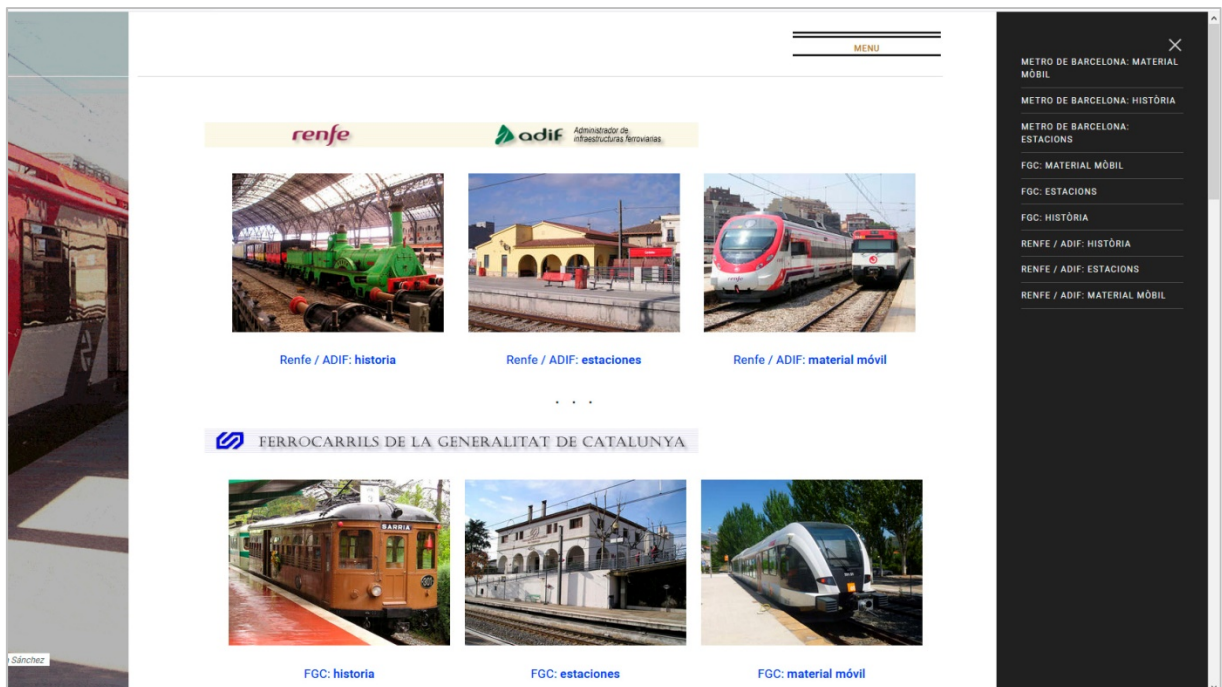


Fig 24: pantalla inicial amb el menú principal desplegat a la dreta

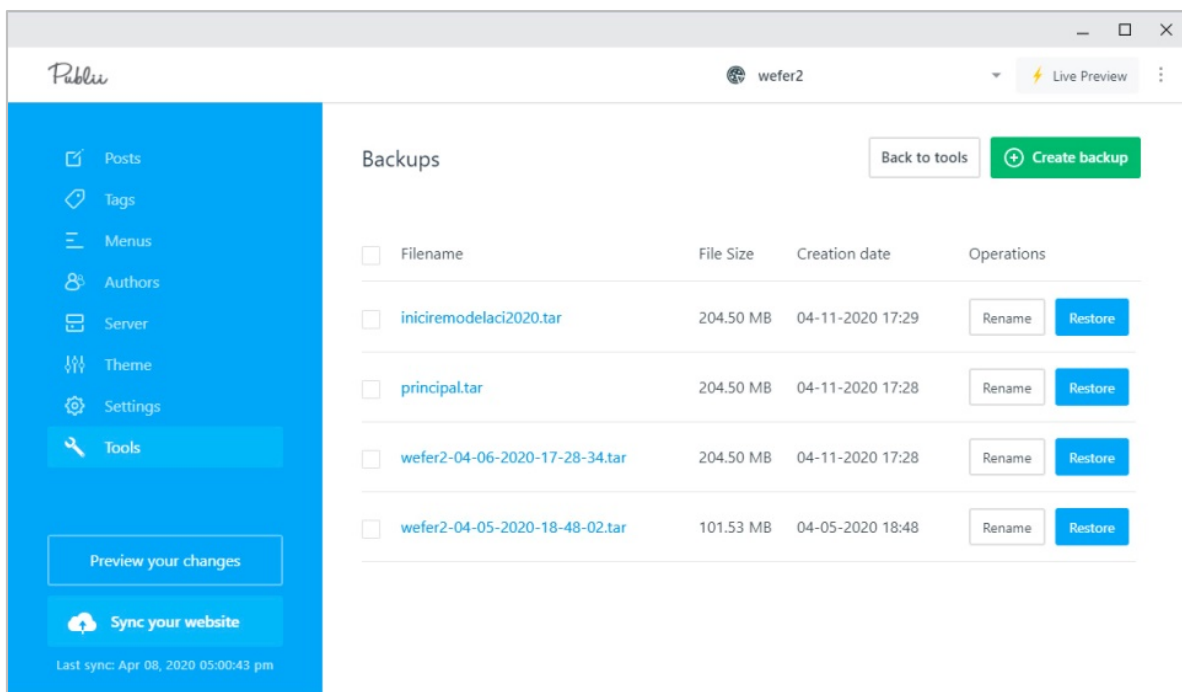


Fig 25: pantalla de backups

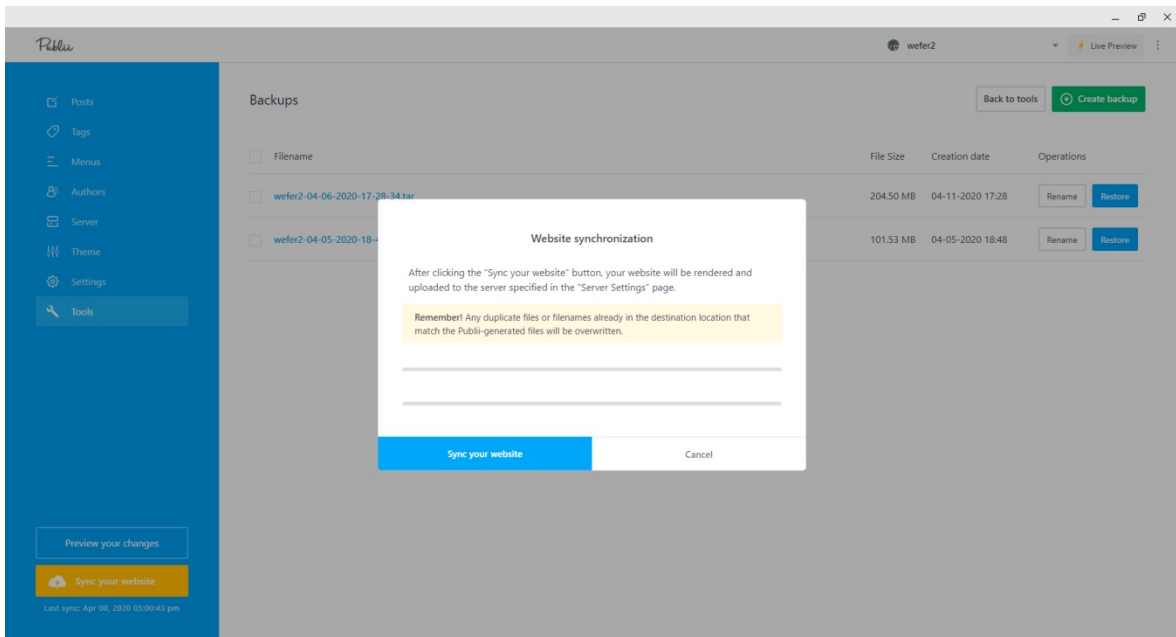


Fig 26: pantalla de sincronització de Publii

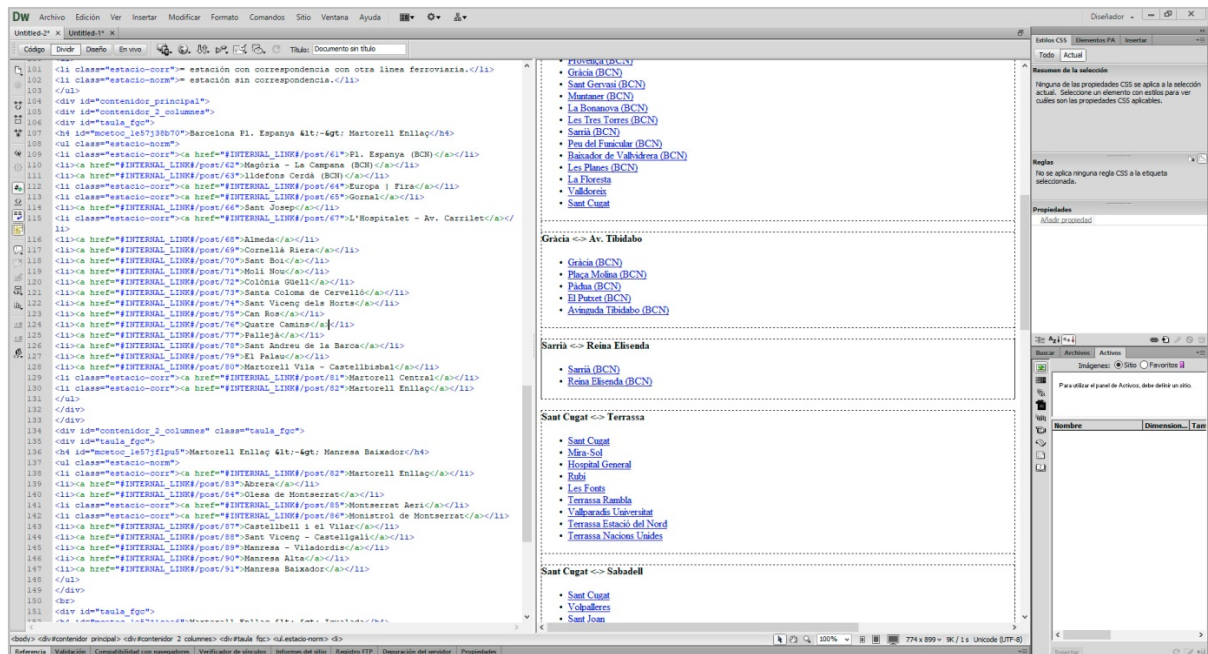


Fig 27: edició HTML amb dremweaver

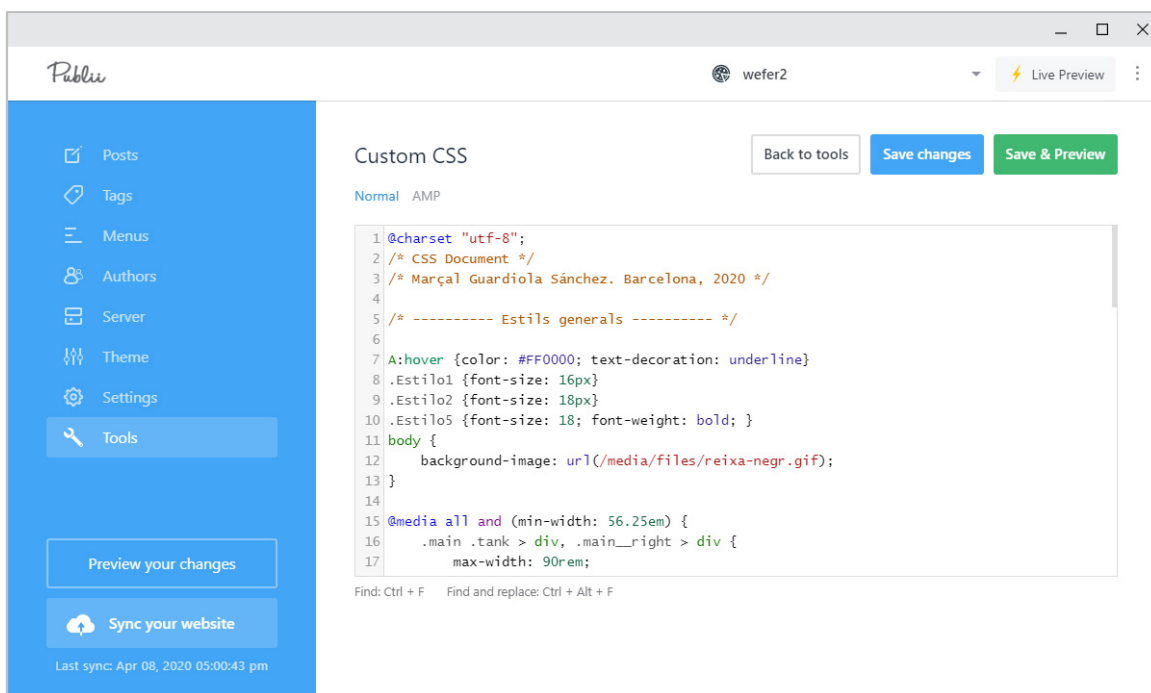


Fig 28: editor CSS intern de Publii

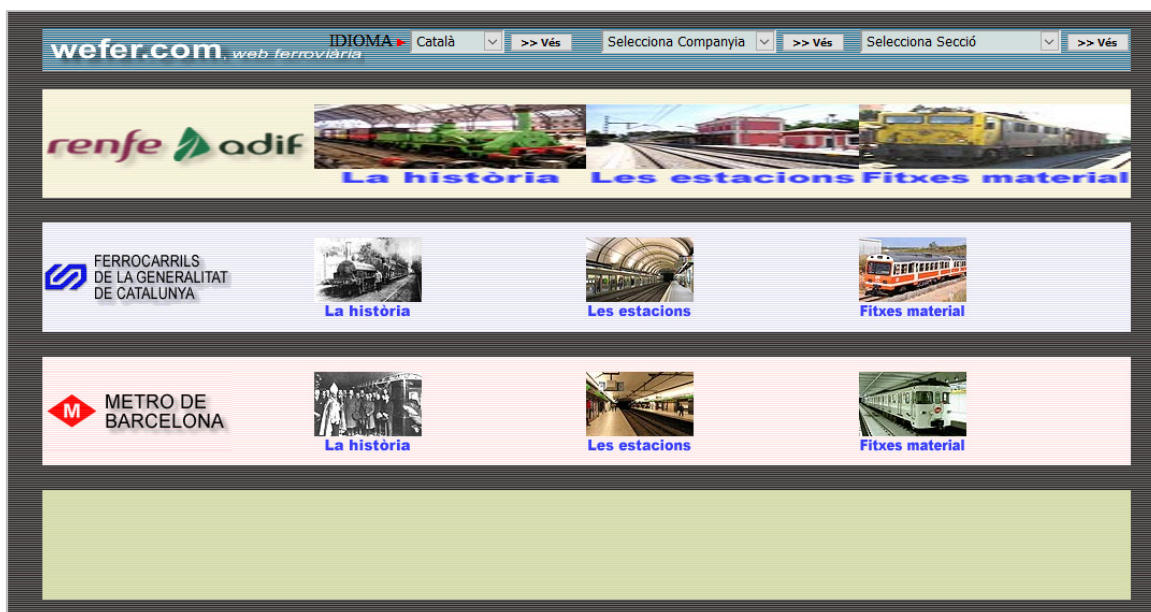


Fig 30: prototip fet en HTML+CSS

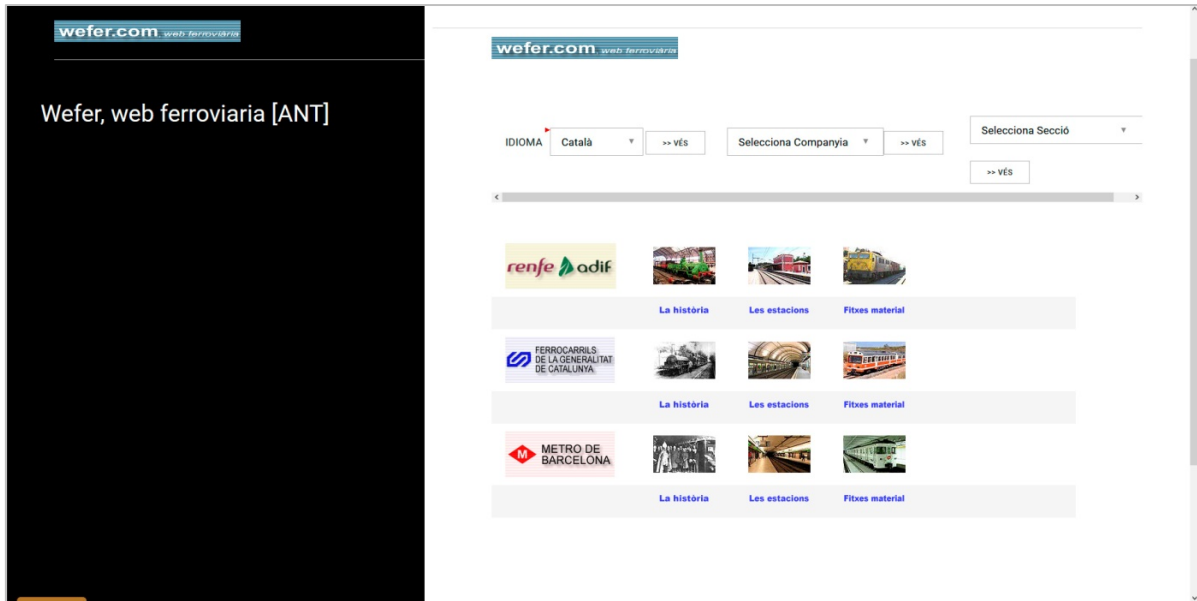


Fig 31: un dels primers prototips fent servir la plantilla Square



Fig 32: aspecte pròxim al definitiu

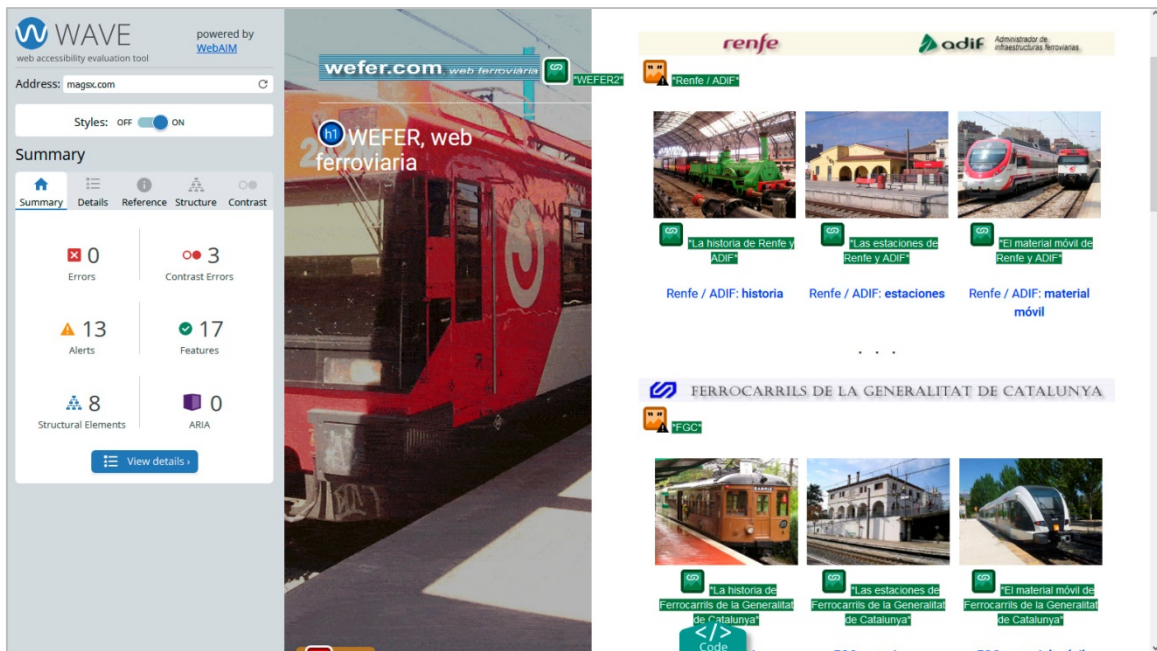


Fig 34: test d'accessibilitat WAVE on es pot comprovar que la pàgina inicial, ja des dels primers moments de desenvolupament, obté 0 errors principals.

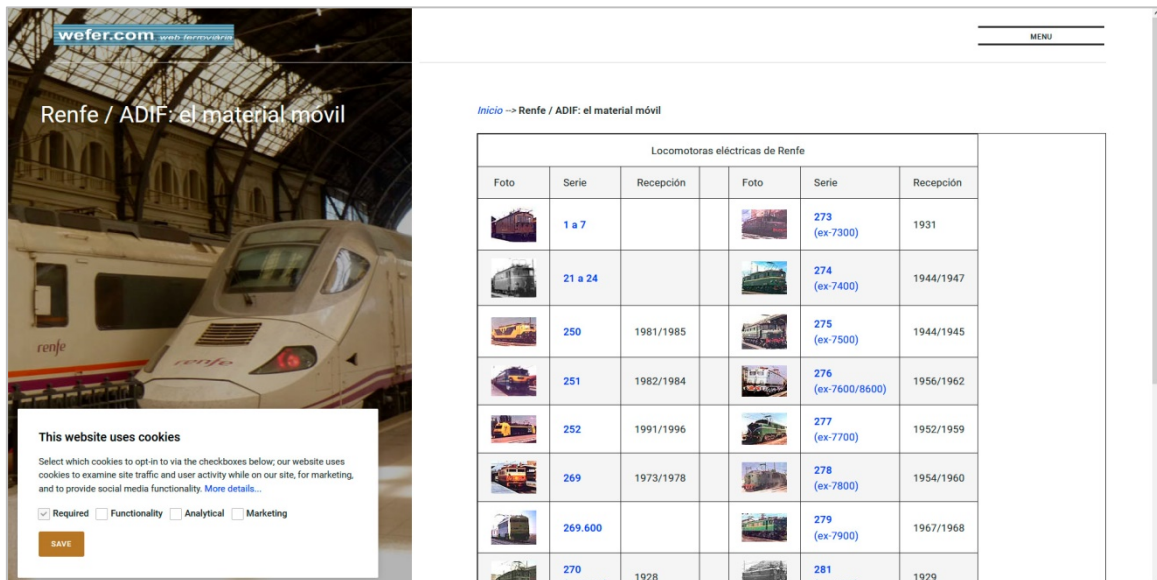


Fig 35: advertiment de "cookies" en compliment del Reglament General de Protecció de Dades (RGPD)



Fig 36: diferències de representació entre previsualització local i definitiva remota

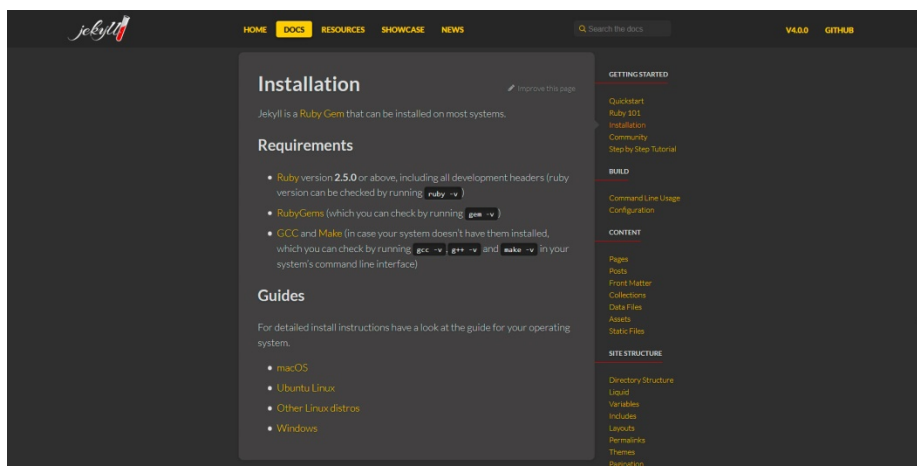


Fig. 37: explicacions de Jekyll sobre els requisits versió per fer-lo funcionar

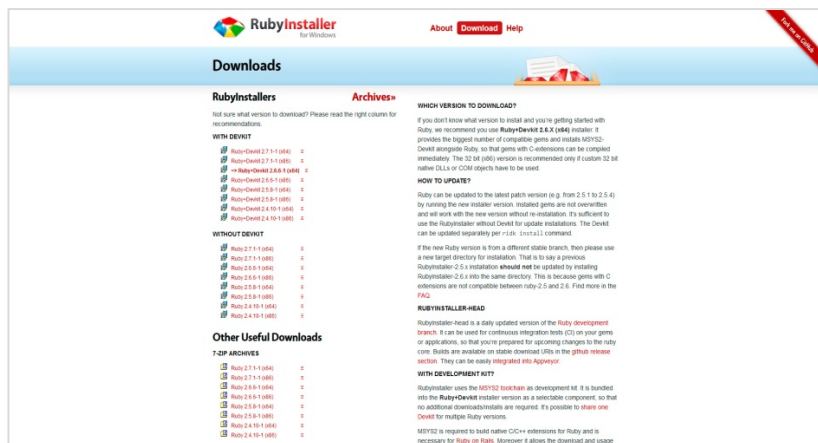


Fig. 38: web de baixada de Ruby: <https://rubyinstaller.org/downloads/>

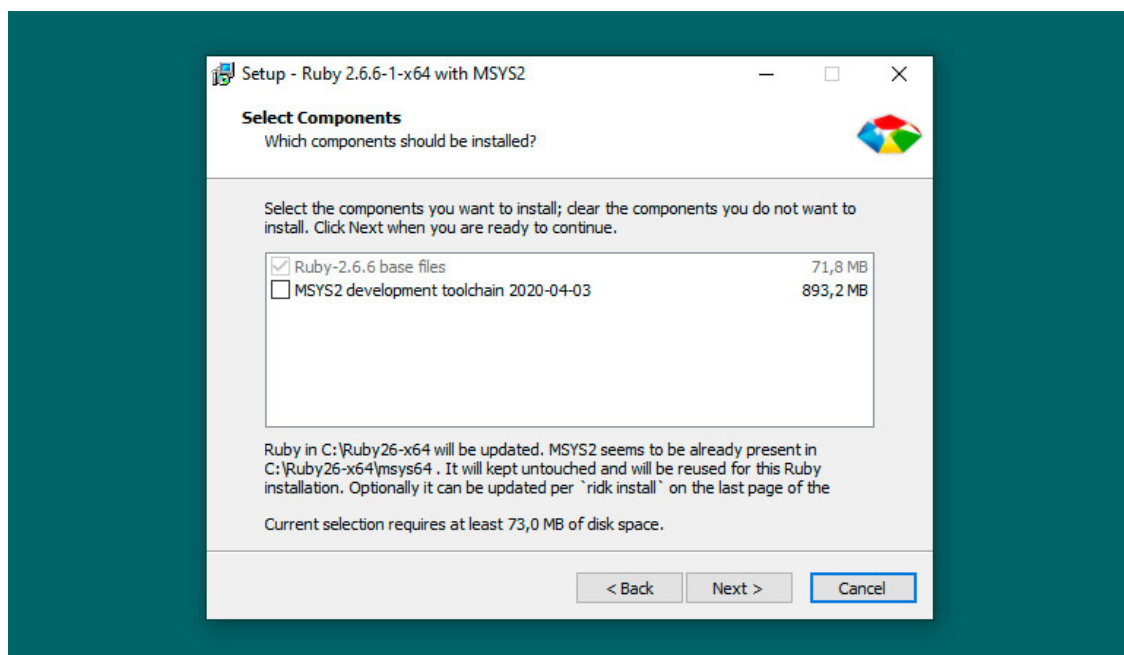


Fig. 39: instal·lació de Ruby a Windows 10

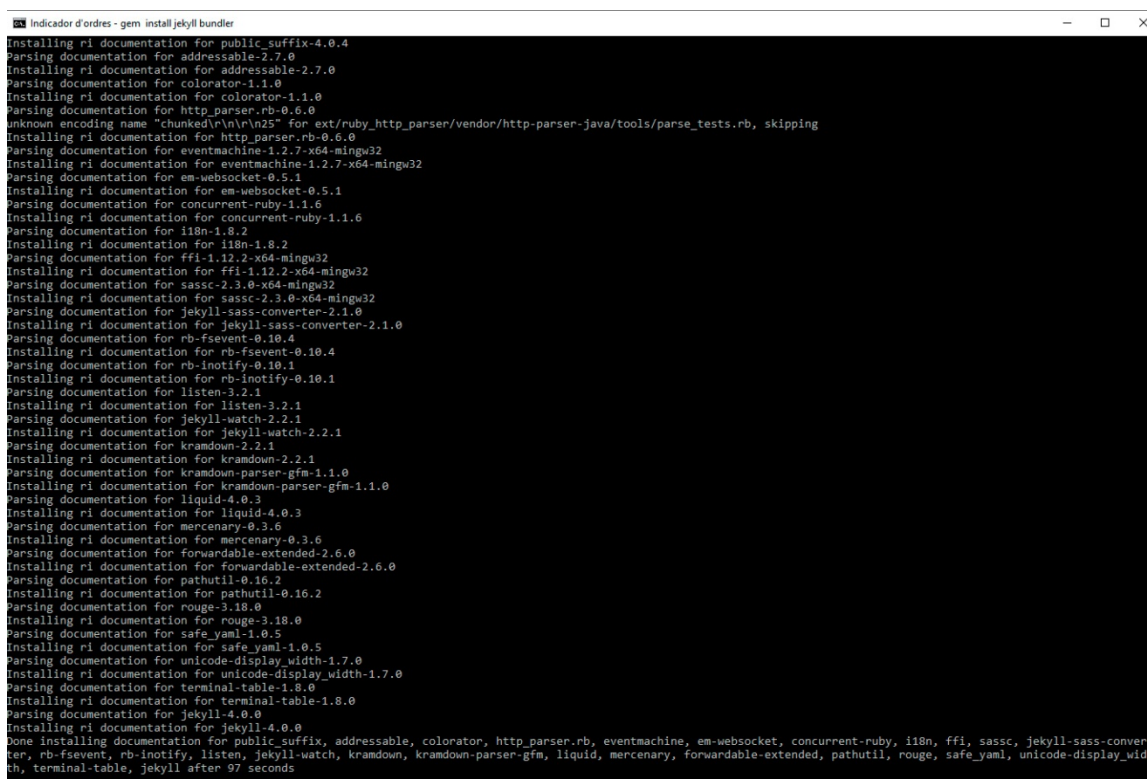


Fig. 40: instal·lació de Jekyll

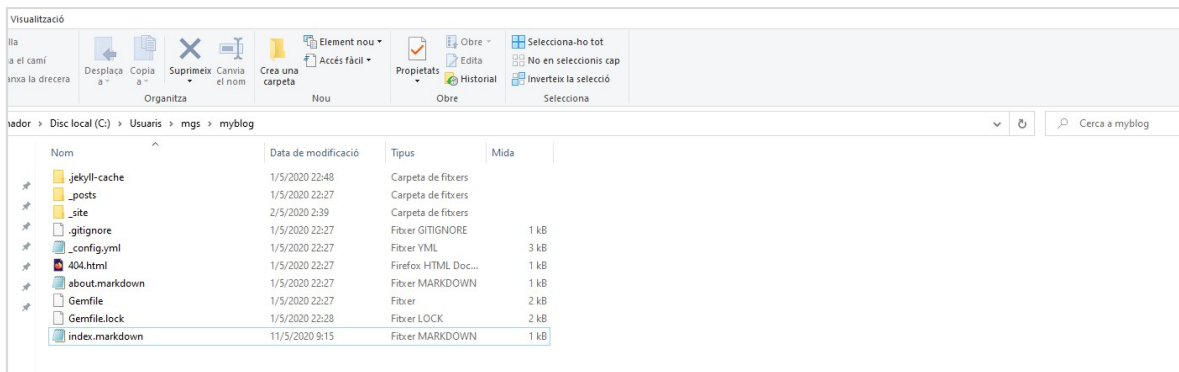


Fig. 41: estructura inicial del nostre blog fet amb Jekyll

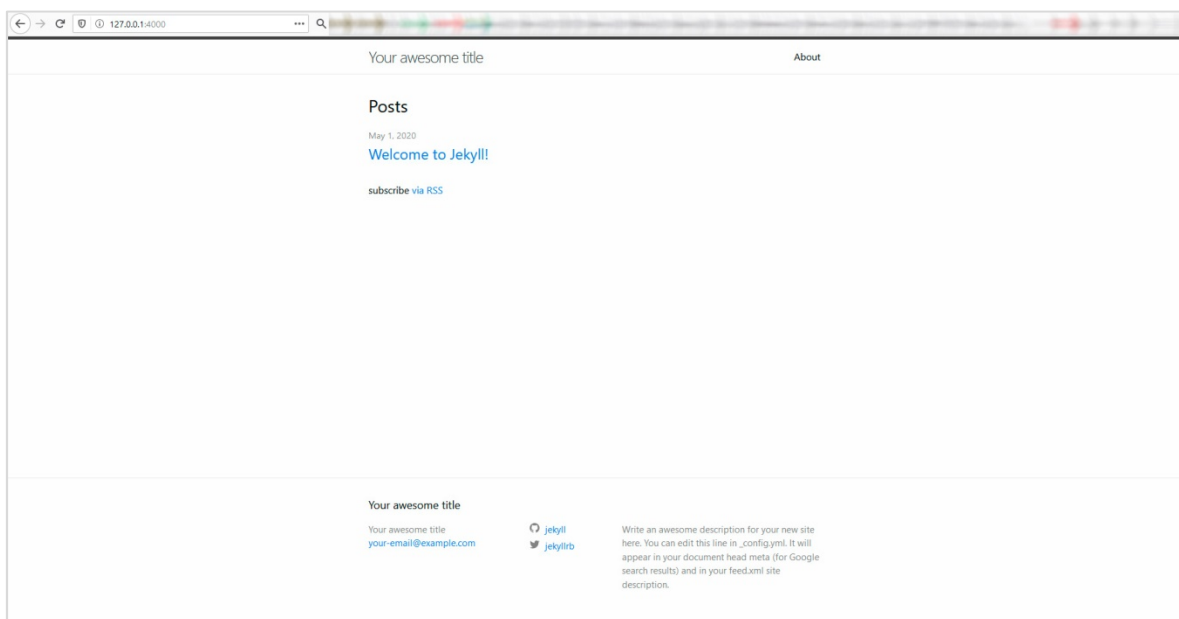


Fig. 42: pàgina per defecte del nostre nou blog amb Jekyll

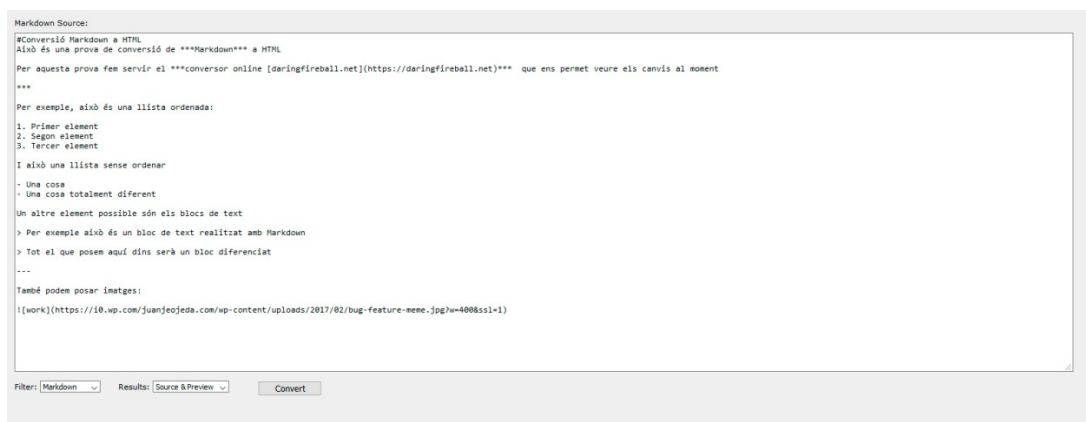


Fig. 43: codi markdown a l'editor markdownguide

```

HTML Source:
<h1>Conversió Markdown a HTML</h1>
<p>Això és una prova de conversió de <strong>em>Markdown</em> a HTML</p>
<p>Per aquesta prova fem servir el <strong>em>conversor online <a href="https://daringfireball.net">daringfireball.net</a></em> que ens permet veure els canvis al moment</p>
<hr />
<p>Per exemple, això és una llista ordenada:</p>
<ol>
<li>Primer element</li>
<li>Segon element</li>
<li>Tercer element</li>
</ol>
<p>I això una llista sense ordenar</p>
<ul>
<li>Una cosa</li>
<li>Una cosa totalment diferent</li>
</ul>
<p>Un altre element possible són els blocs de text</p>
<blockquote>
<p>Per exemple això és un bloc de text realitzat amb Markdown</p>
<p>Tot el que posem aquí dins serà un bloc diferenciat</p>
</blockquote>
<hr />
<p>També podem posar imatges:</p>
<p></p>
  
```

Fig. 44: codi HTML generat a partir del codi markdown

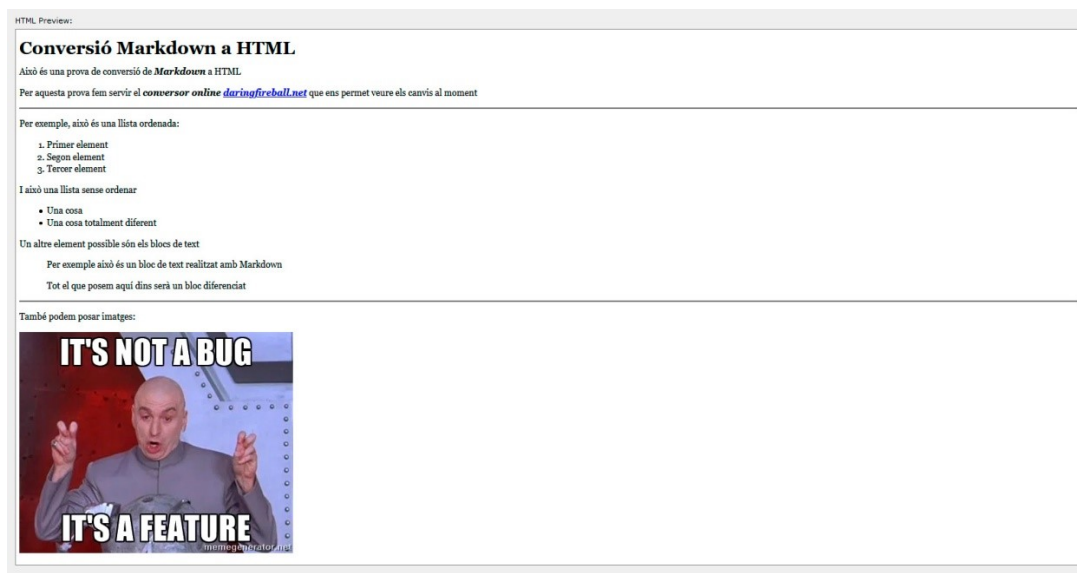


Fig. 45: visualització final del nostre codi markdown al navegador

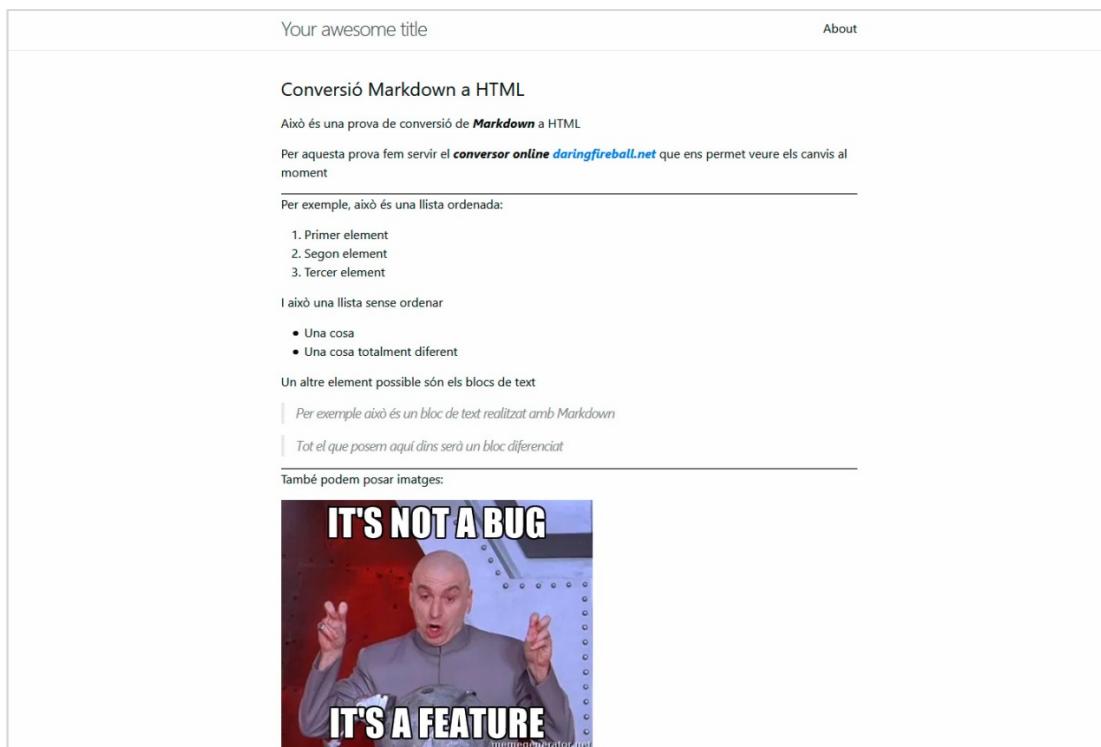


Fig. 46: codi markdown inserta a l'índex del nostre blog amb Jekyll

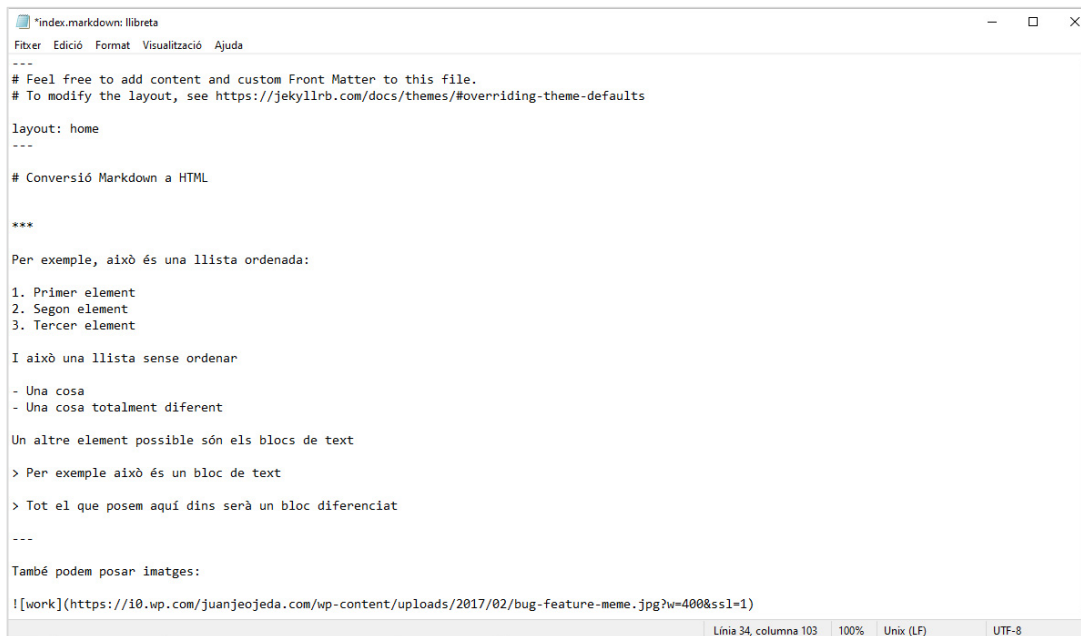


Fig. 47: edició d'un arxiu markdown per a Jekyll (bloc de notes de Windows)

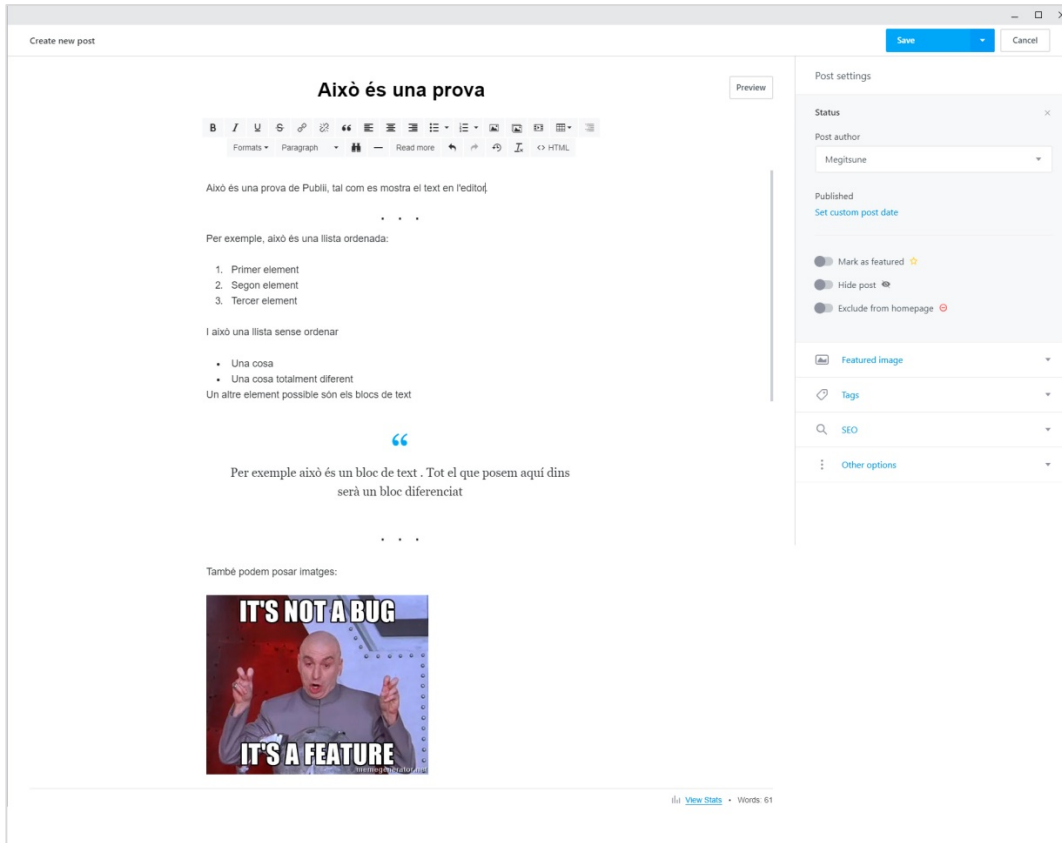


Fig. 48: edició del mateix arxiu anterior fent servir l'editor integrat de Publii

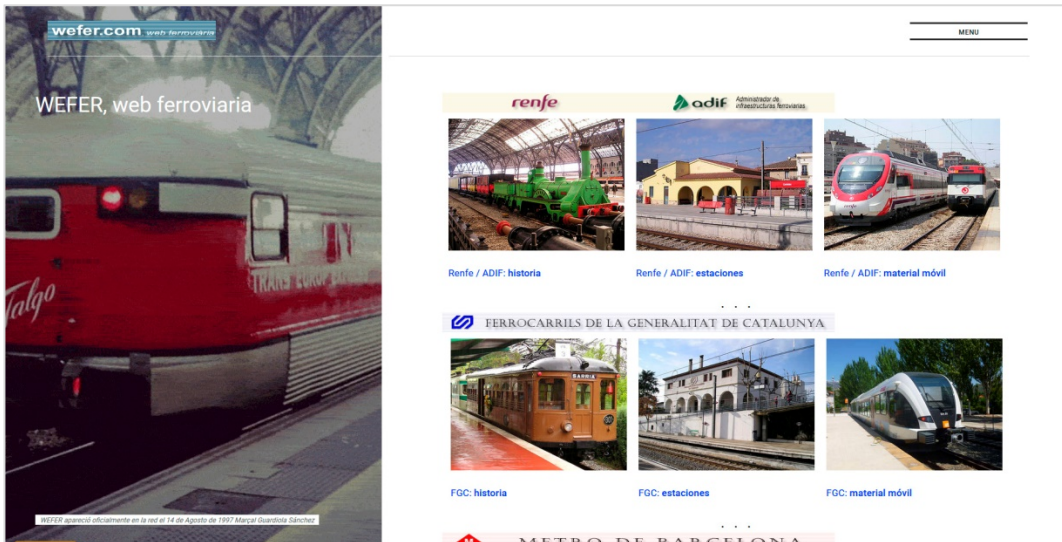


Fig. 49: resultat de la pàgina "index" amb estructura de 3 contenidors horitzontals iguals

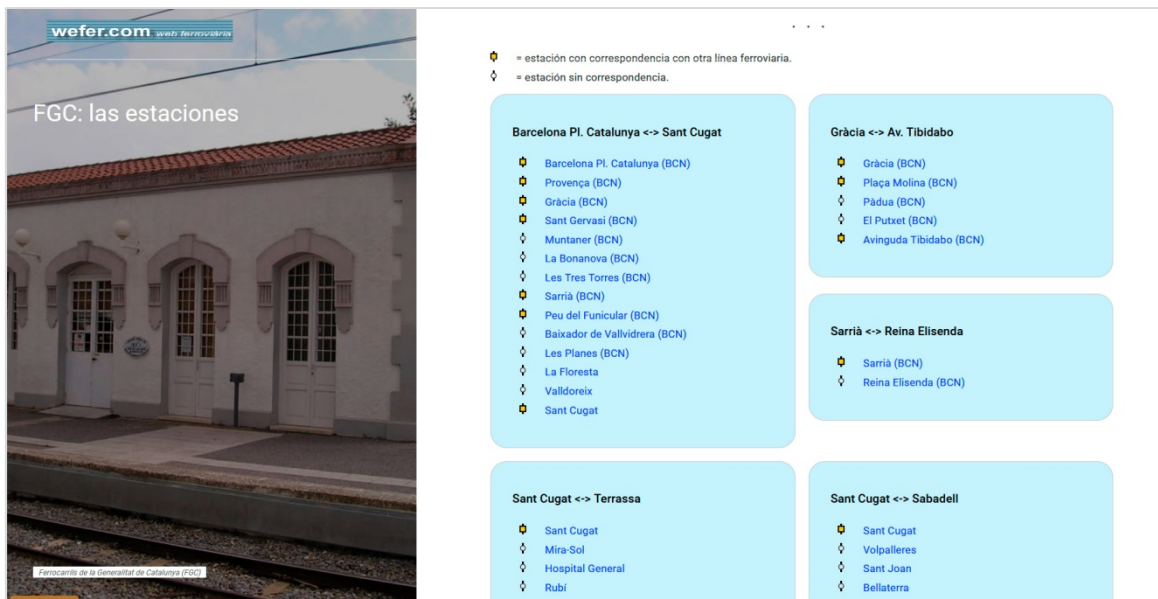


Fig. 50: pàgina "Estacions FGC" amb llistat de dos contenidors i llistes personalitzades

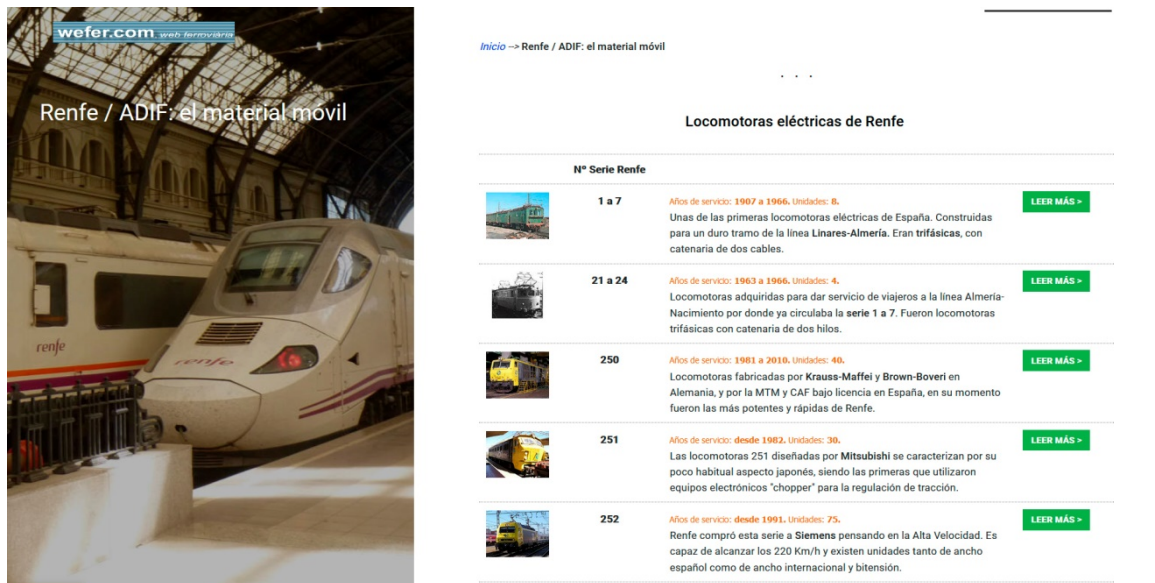


Fig. 51: pàgina "Unitats Renfe" amb llistat horitzontal de 4 contenidors de diferents mides

Locomotoras eléctricas de Renfe

. . .

Nº Serie
Renfe

. . .

	<p>1 a 7 Años de servicio: 1907 a 1966. Unidades: 8. Unas de las primeras locomotoras eléctricas de España. Construidas para un duro tramo de la línea Linares-Almería. Eran trifásicas, con catenaria de dos cables.</p>
	<p>21 a 24 Años de servicio: 1963 a 1966. Unidades: 4. Locomotoras adquiridas para dar servicio de viajeros a la línea Almería-Nacimiento por donde ya circulaba la serie 1 a 7. Fueron locomotoras trifásicas con catenaria de dos hilos.</p>
	<p>250 Años de servicio: 1981 a 2010. Unidades: 40. Locomotoras fabricadas por Krauss-Maffei y Brown-Boveri en Alemania, y por la MTM y CAF bajo licencia en España, en su momento fueron las más</p>




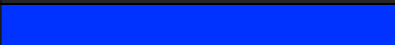



Fig. 52: la mateixa pàgina de la fig. 51, en versió mòbil

Annex 5. Llibre d'estil

En aquests moments el llibre d'estil del lloc web es troba en el seu estadi inicial, però ja podem concretar alguns codis que seran utilitzats en tot el lloc web.

Tipus de lletra: la font principal del lloc web serà l'anomenada ROBOTO amb unes mides que oscil·laran entre 1 i 1,2 REM.

Colors: els colors principals del lloc web seran:

Element	Color	Mostra
Fons	#FFFFFF	
Primari	#a6752e	
Titulars (<i>headings</i>)	#000000	
Text general	#262626	
Enllaços inactius	#0033FF	
Enllaços amb cursos a sobre	#FF0000	
Fons del menú principal	#262626	
Text del menú principal inactiu	#FFFFFF	
Text del menú principal actiu	#FFFB00	

Tots els apartats relacionats amb Renfe tindran el groc com a color predominant:



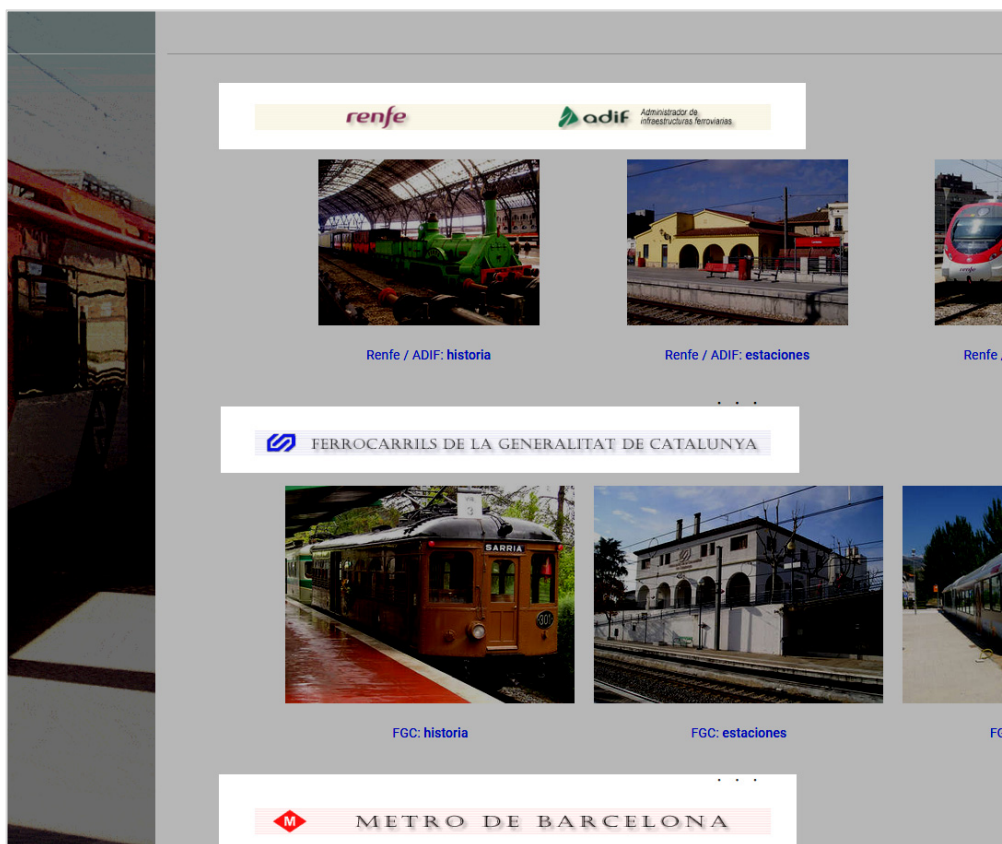
Els apartats dedicats a Ferrocarrils de la Generalitat de Catalunya utilitzaran preferentment el color blau com a fons dels seus elements:



Els apartats on es faci referència a Metro de Barcelona seran preferentment amb imatges i taules amb el fons vermell:



Aquí tenim l'exemple de l'aplicació d'aquets colors en els logotips de la portada:



Annex 6. Glossari

- **Backoffice:** sistema d'administració del lloc web accessible només per als administradors i, per tant, no visible per als usuaris.
- **Backup:** sistema de còpies e seguretat del sistema informàtic, de tal manera que permet recuperar tot el contingut en cas d'error d'actualització, fallada de programari, fallada de maquinari o similar.
- **Breadcrumb:** text que exemplifica la ruta de la pàgina que s'està llegint dins de la jerarquia del lloc web i que serveix per guiar a l'usuari en la seva navegació evitant que es perdi
- **CMS:** sistema web de gestió de continguts que permet elaborar articles, publicar-los i actualitzar-los.
- **Client:** ordinador que demana una informació en el context d'una transacció entre dos sistemes informàtics.
- **Cookie:** petita peça codi que habitualment deixen les pàgines web en els navegadors d'internet per tal d'identificar l'ordinador en concret. S'utilitzen per rastrejar els usuaris o per fer funcionar moltes de les utilitats de personalització dels llocs web.
- **CSS:** full d'estils en cascada. és un sistema per representar els estils que farà servir una pàgina web per tal de ser representada pel navegador. Està íntimament relacional amb el llenguatge HTML.
- **DIV:** objecte contenidor propi del llenguatge HTML.
- **Dreamweaver:** programa per a PC o Mac especialitzat en el disseny de llocs web.
- **FileZilla:** programa per la transferència de fitxers entre dos ordinadors (client i servidor) a internet. Utilitzat habitualment per pujar arxius dels sistemes web.
- **Frame:** funció HTML que permet col·locar més d'una pàgina a la vegada dins la finestra del navegador. és un sistema obsolet.
- **Frontoffice:** part d'un lloc web que veu l'usuari (oposat a backoffice)
- **GDPR:** llei de protecció de dades.
- **HTML:** HyperText Markup Language. És el llenguatge principal de representació de pàgines web de la World Wide Web.
- **Javascript:** llenguatge interpretat que s'utilitza habitualment a les pàgines web per ampliar les prestacions de HTML. S'executa en el costat client, és a dir, en el navegador de l'usuari, en comptes de en el servidor.
- **MySQL:** un dels sistemes més habituals d'emmagatzemar bases de dades a internet. Utilitzat comunament en conjunció amb el llenguatge PHP pr crear pàgines web dinàmiques.
- **PhotoMarks:** programa que permet editar grans quantitats d'imatges de forma ràpida i senzilla. Sobretot útil per posar marques d'aigua de manera massiva.
- **Photoshop:** programa d'edició d'imatges.

- **PHP:** Hypertext Preprocessor. És un llenguatge de programació web que s'executa en el costat servidor i que s'utilitza per crear pàgines dinàmiques.
- **Plantilla:** sistema pel qual podem canviar l'aparença de tot un lloc web de forma senzilla sense necessitat d'anar editant les pàgines una a una. Acostuma a ser un sistema utilitzat en els CMS per facilitar la tasca de disseny.
- **Publii:** programa generador de llocs web estàtics de manera local i amb editors WYSIWYG és a dir, de manera senzilla i assequible per a la majoria d'usuaris.
- **Responsive:** característica d'una pàgina web que fa que el seu disseny s'adapti a la resolució de la pantalla on s'està veient, sigui gran o petit, sense que es presentin problemes de fallades de disseny.
- **SEO:** Search Engine Optimization. Són les tècniques, sempre canviant, que es fan servir per aconseguir que una pàgina web aparegui de manera correcta en els cercadors (Google i similars) i que ho faci en la posició més prioritària possible, per tal d'atraure el màxim d'usuaris possible.
- **Servidor:** ordinador connectat permanentment a internet que serveix pes pàgines web als clients que volen visitar-les.
- **Static Site:** lloc web amb arxius generats amb antelació, és a dir, generats en el moment en que l'administrador ha creat la pàgina web i no pas cada vegada que és demanada per un client.
- **WYSIWYG:** acrònim de "What You See Is What You Get". Es tracta d'una manera de referir-se a tots aquells editors en els quals el format dels canvis es van veient en temps real i no cal esperar a interpretar el codi per veure els resultats del que s'està fent.

Annex 7. Bibliografia

- **Publii Documentation.** [Diverses consultes entre febrer i juny de 2020]. URL: <https://getpublii.com/docs/>
- **A web Coding Blog** [Consulta: febrer de 2020]. URL: <https://www.gotoquiz.com/web-coding/misc/why-are-jsps-so-slow-tomcat-7-vs-apache/>
- **Instituto Tecnológico de Cerro Azul. Programación web.** [Consulta: març de 2020]. URL: <https://nagaaralive0012rh.wixsite.com/mi-pagina-web/12-arquitectura-de-las-aplicaciones-web>
- **Software release life cycle** [Consulta: març de 2020]. URL: https://en.wikipedia.org/wiki/Software_release_life_cycle
- **A List of Static Site Generators for Jamstack Sites.** [Consulta: març i abril de 2020]. URL: <https://www.staticgen.com/>
- **A List of Content Management Systems for Jamstack Sites.** [Consulta: març i abril de 2020]. URL: <https://headlesscms.org/>
- **Why the Publii Open Source CMS is one of the best static website generator.** [Consulta: març de 2020]. URL: <https://medium.com/@bobmitro/why-the-publii-open-source-cms-is-one-of-the-best-static-website-generator-a1b991efa8db>
- **Wikipedia (english). Diversos.** [Consulta: diverses entre febrer i juny de 2020]. URL: <https://en.wikipedia.org/>
- **Wikipedia en español. Diversos.** [Consulta: diverses entre febrer i juny de 2020]. URL: <https://es.wikipedia.org/>
- **Viquipèdia en català. Diversos.** [Consulta: diverses entre febrer i juny de 2020]. URL: <https://ca.wikipedia.org/>
- **Markdown Guide.** [Consulta: abril de 2020]. URL: <https://www.markdownguide.org/basic-syntax/>
- **Ruby installer.** [Consulta: abril de 2020]. URL: <https://rubyinstaller.org/downloads/>
- **Jekyll, referència en línia.** [Consulta: abril de 2020]. URL: <https://jekyllrb.com/>
- **Daring Fireball Dingus (editor Markdown):** [Consulta: abril de 2020]. URL: <https://daringfireball.net/projects/markdown/>
- **Stackoverflow (diverses consultes CSS).** [Consulta: maig de 2019]. URL: <https://stackoverflow.com/questions/4373046/css-control-space-between-bullet-and-li>