

DISSENY I PROGRAMACIÓ D'UNA APLICACIÓ WEB PER LA GESTIÓ D'ESDEVENIMENTS

Raul Ortega López
2n de Batxillerat B
Tutora: Antònia Subirà Ayats

Títol

1. Presentació.....	2
2. Introducció.....	3
2.1. Què cal per a una pàgina web.....	3
2.2. Terminologia i idioma.....	3
2.3. Importància del disseny i l'entorn d'usuari.....	4
2.4. Procés d'aprenentatge dels llenguatges de programació.....	4
3. Procés de desenvolupament i programació.....	5
3.1. Funció i estructura.....	5
3.2. Disseny.....	7
3.3. HTML5.....	16
3.4. CSS+.....	22
3.5. JavaScript.....	27
4. Conclusions i possibles millores.....	33
5. Agraïments.....	35
6. Fonts d'informació.....	36
7. Annexos.....	37
7.1. Accés a un menú amb la versió digital del projecte i la pàgina web. Figura (7.1).....	37
7.2. Accés a una carpeta de drive amb totes les pràctiques de programació realitzades. Figura (7.2).....	37

En aquest treball es realitza el desenvolupament d'una aplicació web per facilitar la celebració d'esdeveniments. Per poder arribar a fer aquesta web es tracta des de l'aprenentatge fins a l'aplicació de disseny i programació web. Tots els passos són explicats de sintètica i entenedora. També es fa ús d'exemples i s'aporten imatges per fer més entenedora l'explicació. Finalment, es fa una reflexió sobre el procés i els resultats d'aquest.

In this work, it is realized the development of a web application to facilitate the celebration of events is carried out. In order to create this website, the learning process, as well as the application of web design and programming, are addressed. All steps are explained in a concise and understandable manner. Examples are also used, and images are provided to make the explanation more comprehensible. Finally, a reflection on the process and its results is made.

1. Presentació

Aquest treball el vaig iniciar a causa de la meva passió per la tecnologia. Sempre m'ha cridat l'atenció. És un camp molt ampli i interessant, que abasta des dels ordinadors o mòbils fins a la creació de materials plàstics. El que la fa més atractiva és la seva versatilitat juntament amb el seu constant canvi. Des de ben petit m'ha cridat aquest món, però sobretot en dos àmbits. El primer és l'enginyeria (sobretot la que se centra en l'arquitectura de components electrònics, com ara els ordinadors i els seus components). El segon, en canvi, és una mica més abstracte, la programació. Aquests dos àmbits per separat són molt interessants, però quan es combinen obren un nou món de possibilitats. En aquest cas, però, vaig decidir centrar-me en un, el que més em crida, la programació web. Fent aquest treball pretenc aprendre les bases de la informàtica necessàries a l'hora de pensar, dissenyar i crear una pàgina web des de zero. Vull aprendre quins són els passos a seguir, quins llenguatges són necessaris i fins i tot quant de temps cal dedicar a un projecte com aquest. El que m'ha empès a embrancar-me en aquest projecte és l'aspiració que tinc d'estudiar enginyeria informàtica per dedicar-me a la programació. Aquest treball em pot donar una bona base per emprendre aquest repte.. Començar amb els coneixements que puc adquirir a l'hora d'aquest treball pot ser un avantatge a l'hora de fer algunes assignatures. Tot i que no tot és per la carrera, també ho faig tal com he mencionat perquè m'apassiona. Fer aquest treball em permet aprendre sobre un món que per mi és desconegut i em permet endinsar-hi a fons.

Per tal de facilitar-me les coses em vaig fixar alguns requisits mínims de com volia que acabés sent la meva pàgina web. La intenció era que fos una web capaç de crear esdeveniments, on poguéssim emmagatzemar productes, que es poguessin marcar com adquirits, i que fos possible compartir les llistes amb altres persones. Això facilitaria molt l'organització dels esdeveniments, fent-los més divertits i alleugerant la feina de l'organitzador.

2. Introducció

2.1. Què cal per a una pàgina web

Tot i que fer una web en primera instància pugui semblar simple, no ho és. Programar és complicat.

Una pàgina web no es limita a la programació de codi. Aquesta és una de les parts més important lògicament, però n'hi ha altres de rellevants.

Per començar hem de saber que és el que volem fer. Per tant, hem de tenir una idea clara sobre el tema de la nostra pàgina. En segon lloc, hem de dissenyar i maquetar aquesta pàgina. En tercer lloc, hem de saber programar (HTML, CSS i JavaScript). Finalment, serà necessari comprar un domini per poder penjar aquesta pàgina. Tal com es mostra a la figura (2.1)

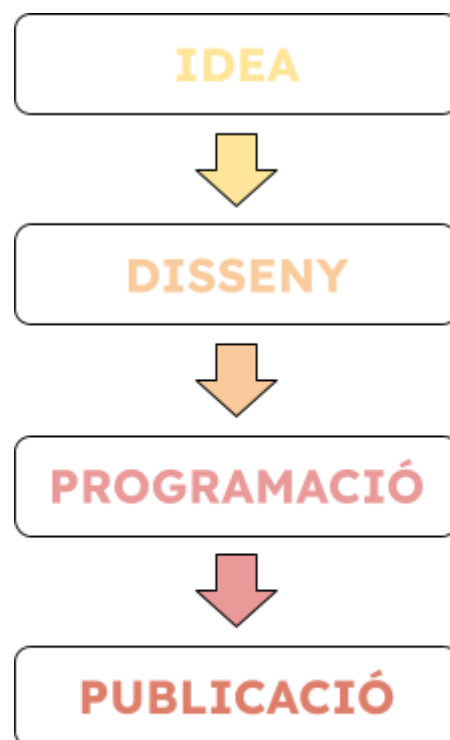


Figura 2.1 Esquema per crear una pàgina web

2.2. Terminologia i idioma

Quant a terminologia, és necessari l'ús de termes concrets, ja que, faciliten molt l'explicació del procés de programació i l'explicació del funcionament. Per aquest motiu tota la terminologia que aparegui en el treball serà esmentada prèviament, o bé serà explicada de forma breu.

L'idioma és un aspecte molt important en la creació de pàgines web. Aquest definirà a quin públic està orientada la pàgina i haurem d'escollir-lo segons els nostres objectius. En aquest cas la pàgina ha estat escrita en anglès per tal d'arribar al màxim de públic possible i l'anglès és la llengua estàndard. Però, com ja he mencionat aquest projecte seguirà viu després de l'entrega i presentació del meu

treball de recerca. Per tant, continuarà millorant i probablement hi afegiré una funció de canvi de llenguatge, la qual encara no ha estat implementada per manca de temps.

2.3. Importància del disseny i l'entorn d'usuari

La majoria de les persones releguen l'apartat estètic de les pàgines web a un segon pla, sense saber que és una de les parts més rellevants. Tot i tenir un bon codi si el treball de disseny i entorn d'usuari no estan polits serà complicat i incòmode fer servir la web. Fent que sigui poc intuïtiva, lenta i confusa. Tot això es pot evitar fent ús d'una estètica simple, i un entorn d'usuari ben treballat. Per tant, una web no serà millor per tenir més funcions, sinó perquè les que tingui siguin senzilles i fàcils d'utilitzar per a l'usuari. Si seguim aquests criteris també farà la nostra web molt més accessible a qualsevol públic.

2.4. Procés d'aprenentatge dels llenguatges de programació

Aquesta, és la part més subjectiva del treball. Hi ha moltes maneres d'aprendre un llenguatge de programació. Per tant, no dic que la forma explicada a continuació sigui la correcta, sinó la que jo he usat m'ha funcionat.

Com que en iniciar el treball mai havia fet programació amb HTML, CSS o JavaScript vaig començar fent una cerca d'informació sobre aquests llenguatges. Tot i que n'hi ha molta en un inici no em va ser massa útil perquè la majoria està dedicada a les persones que ja tenen coneixements en la matèria. Un altre dels problemes que em vaig trobar és que com són elements complexos, sobretot JavaScript, la majoria dels cursos eren de mesos, i no m'ho podia permetre. Per tant, finalment vaig decidir mirar vídeos de YouTube que expliquessin com funcionaven els llenguatges que havia d'aprendre i ho exemplifiquessin. Després n'extreia les parts més útils i les feia servir en el meu codi. Finalment, vaig posar en pràctica tots aquests coneixements en els programes que es poden veure en els annexos.

3. Procés de desenvolupament i programació

3.1. Funció i estructura

Tenir clar com funciona i els objectius bàsics és clau a l'hora de realitzar una bona web. Per això el primer que cal és definir uns objectius per poder treballar sobre aquests. Els que jo em vaig marcar van ser els següents.

- Poder crear llistes de productes necessaris per esdeveniments.
- Possibilitat de crear tants esdeveniments com siguin necessaris.
- Els productes podran ser marcats com adquirits i desmarcats en qualsevol moment.
- Els productes marcats no es podran eliminar fins a ser desmarcats.
- Tot serà en línia, per tal de poder veure les actualitzacions de forma immediata.
- Els esdeveniments es podran compartir amb altres usuaris de la web mitjançant el correu electrònic o bé el nom d'usuari.
- Es podrà iniciar sessió o registrar-se amb de forma ràpida amb Google o Apple
- La web estarà optimitzada tant per mòbil com per ordinador.

Un cop vaig tenir clars els meus objectius vaig investigar com funcionaven les pàgines web i com es podien comunicar amb una base de dades per gestionar la informació. L'estructura de la comunicació és la que es pot veure a la figura (3.1). Quan tenim qualsevol interacció amb la web aquesta es comunica amb el servidor, on hi ha tot el codi. La informació és gestionada per un programa escrit amb l'HTML i seguidament torna a ser gestionada és per un altre de JavaScript, que mitjançant les api ¹farà que arribi a la base de dades (BBDD MySQL), on cercarà que ha de fer en rebre la informació de l'acció de l'usuari. Un cop trobi la resposta la informació farà el recorregut invers fins a tornar a la web, on la mostrarà a l'usuari. En cas de no trobar la informació la base de dades tornara un error amb un número el qual indica de quin tipus d'error es tracta.

¹ API. És una interfície de processament d'aplicacions entre un servidor web i un navegador web

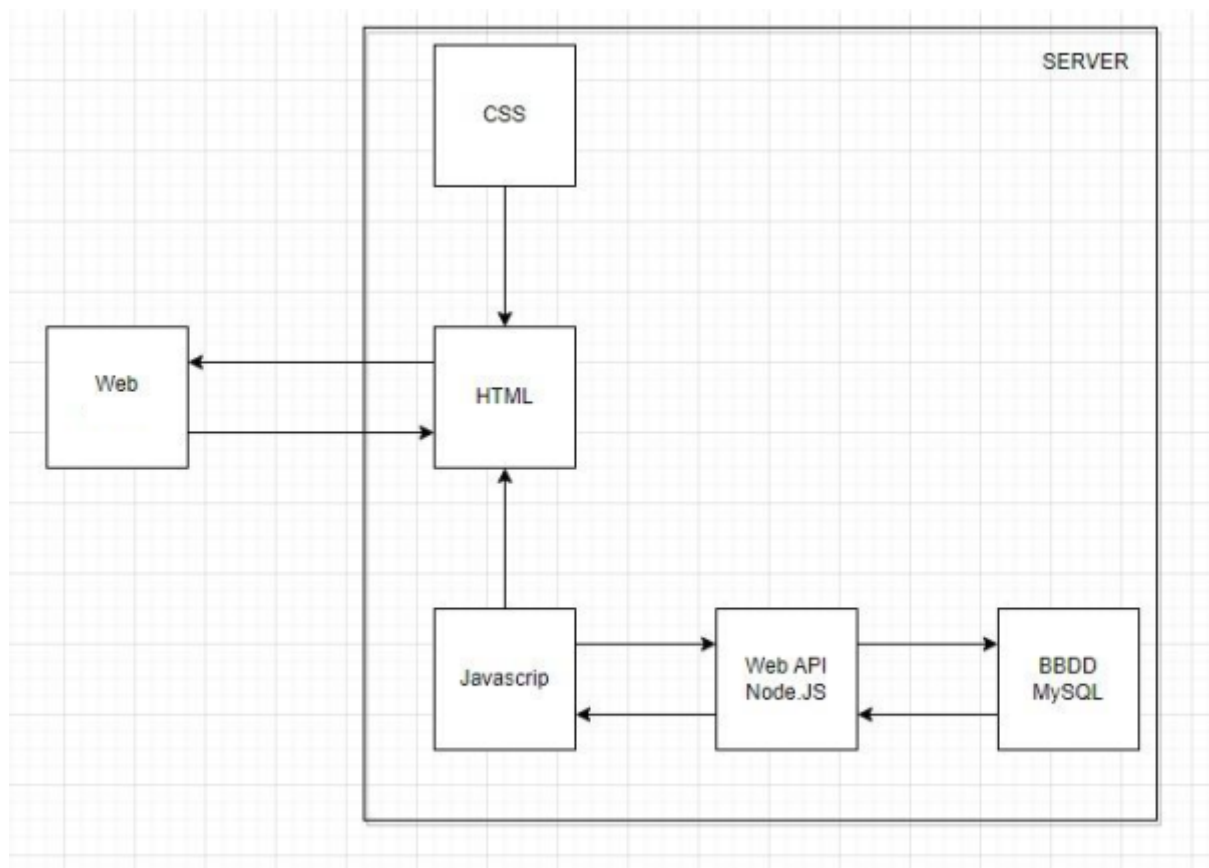


Figura 3.1 Esquema del funcionament de la pàgina web

L'esquema de la figura (3.2) ens mostra com es transmet la informació dins la base de dades. Els productes estan arrelats a les llistes. Les llistes estan connectades als usuaris, tant al propietari, com als que està compartida i aquests també estan connectats entre si. Els usuaris estan relacionats perquè la llista està compartida

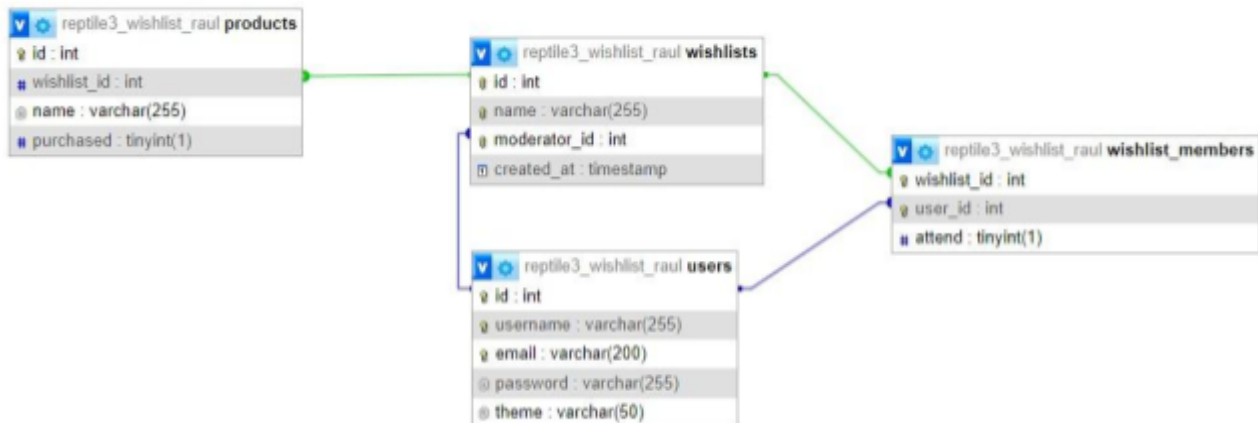


Figura 3.2 Esquema de la base de dades

3.2. Disseny

El primer pas a l'hora de crear una pàgina web un cop tenim clar de què tractarà és el disseny. És important definir-lo des d'un inici per així poder treballar d'una forma correcta i seguint unes pautes. Si no el definíssim seria pràcticament impossible desenvolupar una pàgina complexa i que tot tingues sentit i concordança en l'àmbit estètic.

El primer pas a seguir és definir una idea principal molt esquemàtica i intentar plasmar-la d'alguna manera. Una bona idea és fer un dibuix ràpid de com vols que sigui el teu lloc web. Plasmarem totes les pàgines que ha de tenir i especificarem les funcionalitats principals de la web (quina serà la relació entre pàgines, quina forma ha de tenir la nostra web, quin serà el posicionament dels elements principals a les diferents pàgines...). No cal que sigui idèntic al resultat final, però servirà per acotar idees i fer el següent pas molt més senzill.

Un cop tenim una idea inicial i un esbós hem de començar a concretar quin serà² l'estil de la nostra pàgina. Aquí sí que haurem d'acotar al màxim i fer que el disseny sigui tal com volem que quedi la web final. Per dur-ho a terme podrem fer ús de webs especialitzades en disseny gràfic. Aquestes ens facilitaran la creació d'elements i el seu posicionament. També haurem de començar a definir els noms dels elements (títols, botons, "modals"...) i la gamma cromàtica que tindrà la nostra web.

Vaig començar fent un esquema a mà alçada fent ús de les tauletes gràfiques. Aviat vaig veure que no era gaire pràctic, pel fet que no tinc experiència en l'ús d'aquestes eines i trigava molt temps a fer els dissenys. A més no eren gaire específics. Com que no tenia experiència en el disseny de l'estructura de les webs anava a cegues dibuixant el que pensava que es podria ajustar millor al meu projecte.



Figura 3.3 Esbós a mà alçada de la web

² "modals": finestres emergents, que com veurem més endavant, proporcionen informació o requereixen alguna acció.

Finalment, però vaig decidir provar de passar al segon pas sense haver acabat el primer (a causa de les complicacions mencionades). En aquest cas em vaig decantar per Figma, una web de disseny gràfic molt completa i realment utilitzada pels dissenyadors professionals. Aquesta web ens permet crear pàgines que podrem modificar afegint-hi: imatges, text, formes... Després de fer una cerca de colors vaig decidir que empraria una gamma de colors: grocs, blancs, negres i grisos. Un cop escollida l'eina que volia fer servir i els colors principals vaig iniciar el procés de desenvolupament gràfic

El primer que vaig decidir dissenyar va ser el conjunt de "login", figura (3.4), i "register", figura (3.5), pel fet que són dues de les parts més estàndard. Els hi pots donar diferents estils, però en essència sempre han de complir amb el mateix comès.

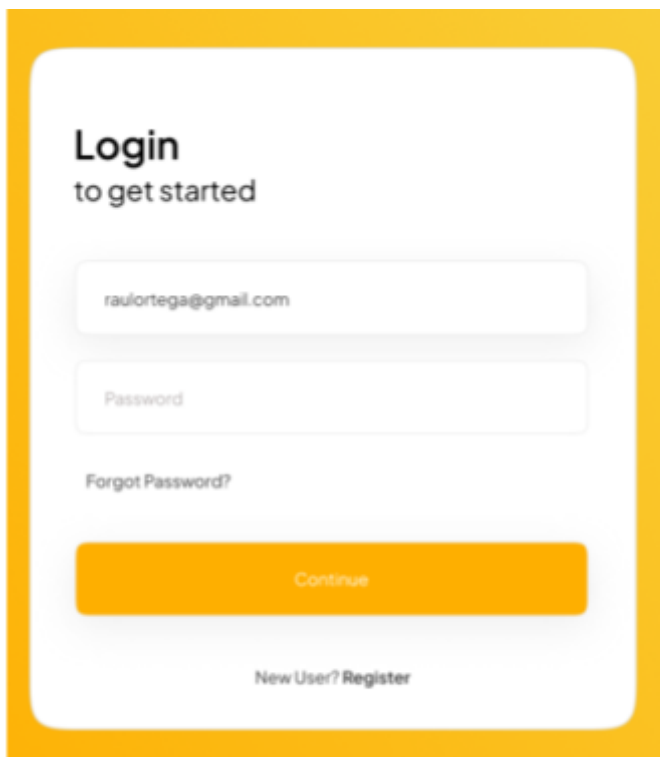
A login form design with a yellow border. The title "Login to get started" is at the top left. Below it are two input fields: the first contains the email "raulortega@gmail.com" and the second is labeled "Password". A link "Forgot Password?" is positioned below the password field. A large yellow "Continue" button is at the bottom, with a link "New User? Register" centered below it.

Figura 3.4

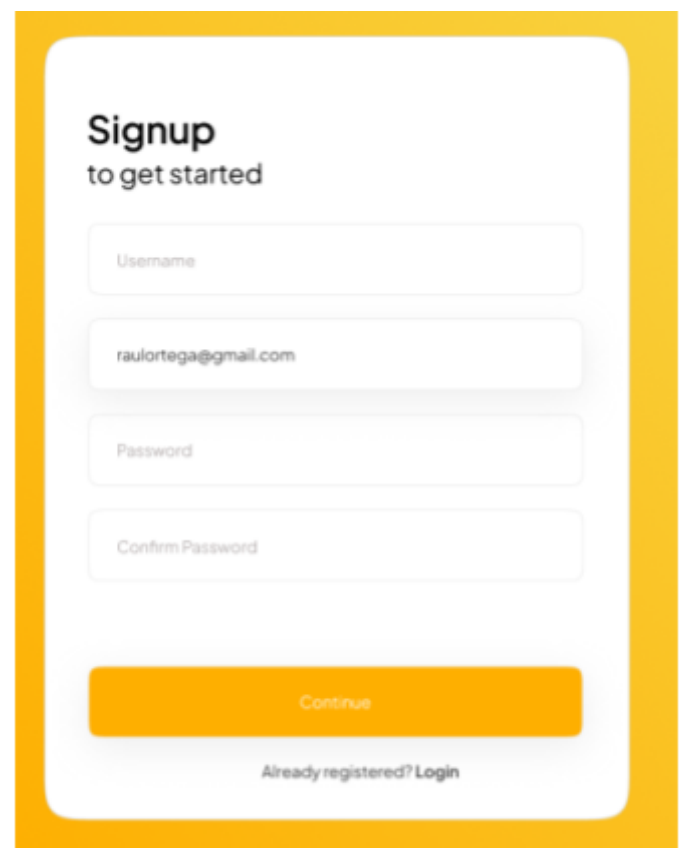
A signup form design with a yellow border. The title "Signup to get started" is at the top left. Below it are three input fields: the first contains the email "raulortega@gmail.com", the second is labeled "Password", and the third is labeled "Confirm Password". A large yellow "Continue" button is at the bottom, with a link "Already registered? Login" centered below it.

Figura 3.5

Un cop fets els dos apartats més generals de la web vaig iniciar la pàgina més important, la (“home page” - pàgina principal), figura (3.6). En aquest cas tots els estils ja han de ser personals i pensats per un mateix, no hi ha cap pauta a seguir. Vaig decidir seguir un estil minimalista, presentant en pantalla només allò que és necessari. D’aquesta manera la web va quedar molt polida i intuïtiva.

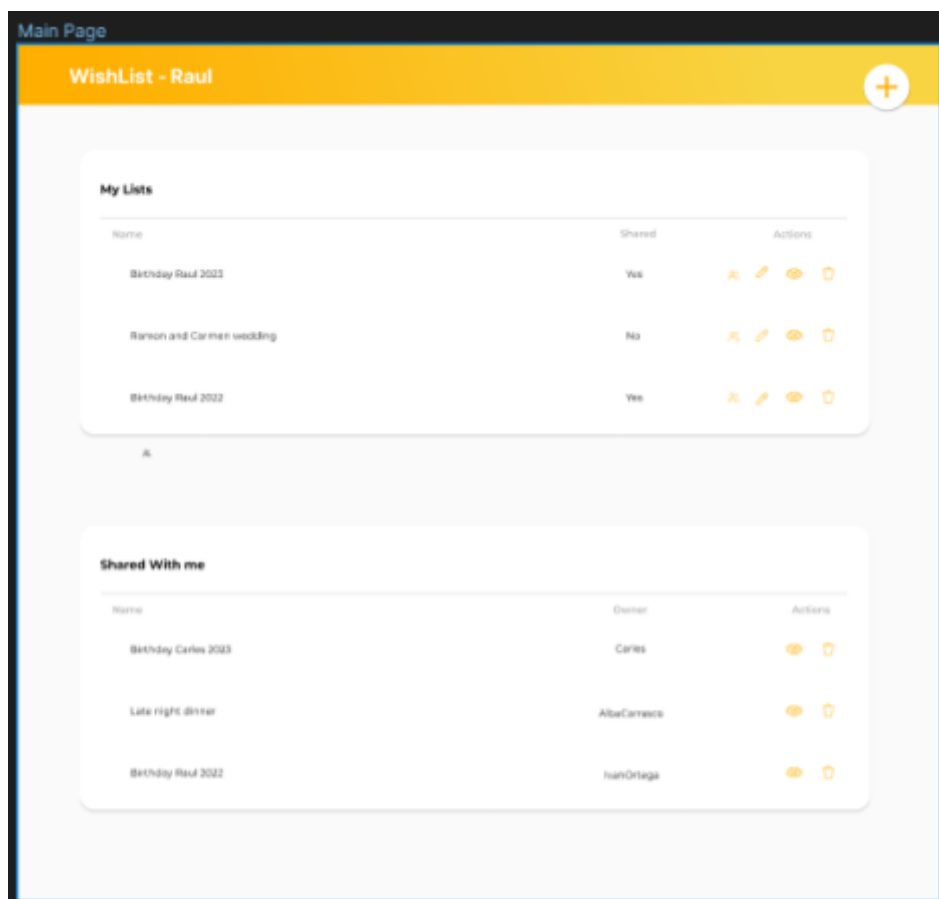


Figura 3.6.1. Disseny de la pàgina principal de la web

El primer element necessari és un botó per afegir-hi els esdeveniments. El podem veure a la imatge a la part superior dreta, és un símbol de suma. En clicar-hi, tal com es mostra a la figura (3.7), demana el nom d’una llista i que confirmem si la volem crear o cancel·lem l’operació.

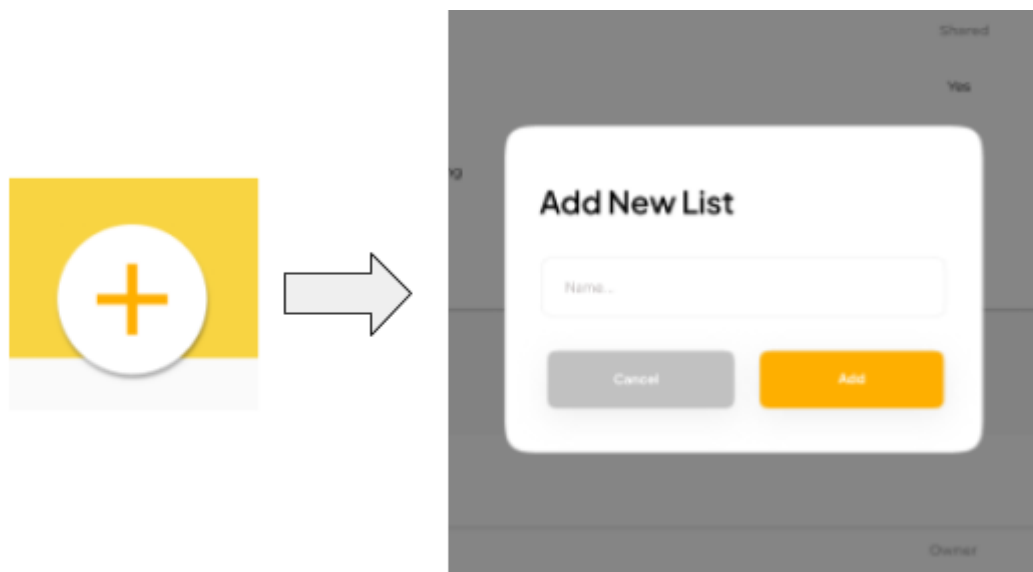


Figura 3.7 Acció en clicar el botó “+”

En segon lloc, a la part superior esquerra hi trobem el nom de la web, que en totes les altres pàgines ens servirà per tornar a la d'inici, i el nom d'usuari el qual en clicar-lo ens durà en un menú. Aquest va ser un dels últims a ser dissenyat. Com veiem a la figura (3.8) respecta l'estil de la web i ens mostra informació bàsica de l'usuari (nom i adreça electrònica) i el botó de “log out”, que ens permetrà sortir de la pàgina tancant la sessió.



Figura 3.8 Acció en clicar al nom d'usuari

En últim lloc, si ens fixem en la zona de les llistes hi podem trobar tres elements dividits en tres columnes:

1. El nom: On trobarem el nom que nosaltres hàgim decidit per aquella llista
2. L'estat de l'esdeveniment quant ha compartit: Aquest apartat ens indicarà si hem compartit l'esdeveniment o no.
3. I finalment les accions: Aquest apartat és el més rellevant. El que es mostra a la figura (3.9) són quatre botons i aquests són els que ens permetran interactuar amb els esdeveniments. Tots ens obriran un "modal" en clicar-hi menys el segon, el qual ens redirigirà a la pàgina de l'esdeveniment.



Figura 3.9 Botons d'acció de la pàgina principal de la web

Com podem veure el primer que trobem és la icona per visionar els assistents al nostre esdeveniment. Un cop hi cliquem com se'ns mostra a la imatge () podem veure els participants, si hi assistiran o no i un formulari per indicar la nostra assistència. Si la llista no està compartida al web no ens mostrarà la icona.

La segona icona que hi veiem és la d'edició. Aquesta ens permetrà accedir a la pàgina on visualitzarem el contingut de l'esdeveniment i el podrem modificar. És l'única que no desplega un modal i, per tant, l'esmentarem més endavant.

En tercer lloc, trobem un ull, el qual és el botó de visionament i ens indica els productes que no estan marcats com adquirits al nostre esdeveniment, figura (3.10). Això ens permet estalviar temps a l'hora de saber els elements que queden sense marcar i no haver de buscar-los entre tots els altres. En cas que no hi hagués productes sense marcar la pàgina ho detectarà automàticament i no ens el mostrarà.

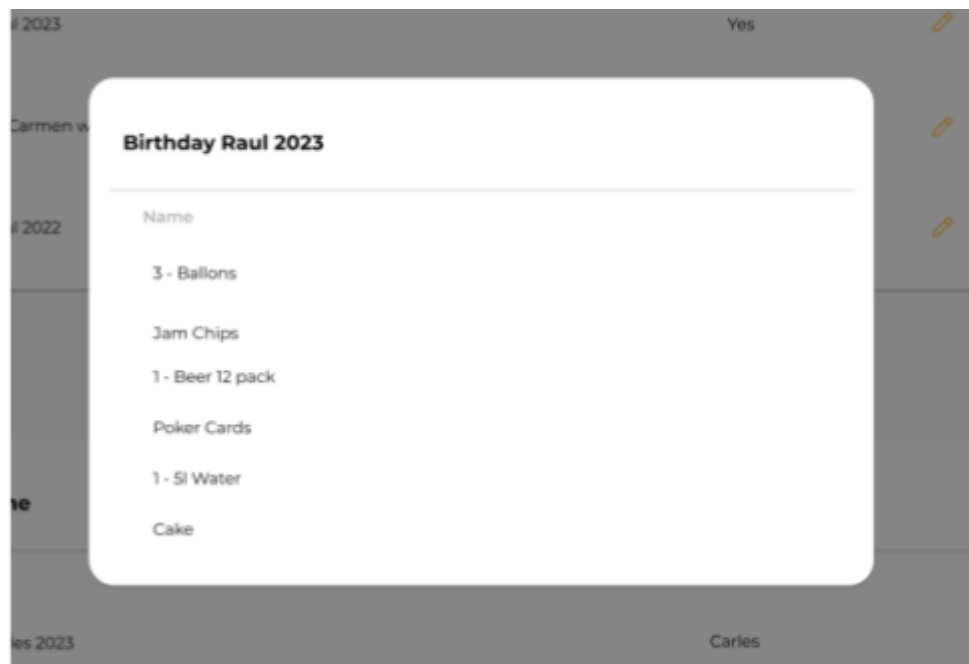


Figura 3.10 Vista dels productes que no estan marcats com adquirits

Finalment, trobem una paperera que ens servirà per eliminar l'esdeveniment en cas que ja hagi acabat o ja no hi vulguem participar. Figura (3.11)

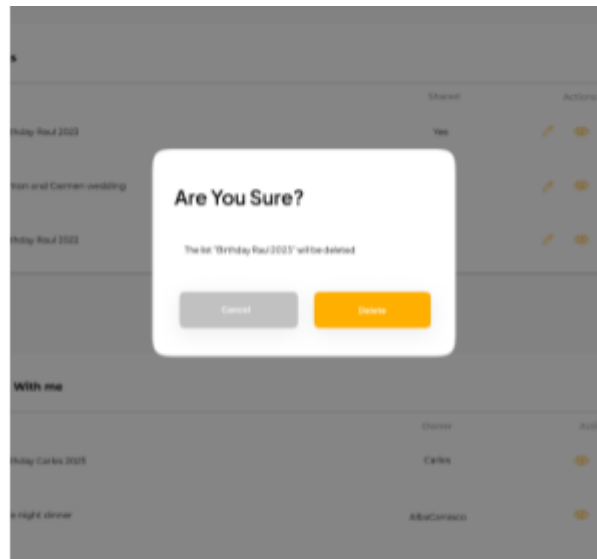


Figura 3.11 Questionari de confirmació per eliminar un esdeveniment

Un cop acabada la pàgina principal i els “modals” vaig començar el disseny de la pàgina on visualitzaríem els esdeveniments de forma individual i els seus productes, figura (3.). Per accedir-hi s’ha de clicar sobre el botó en forma de llapis, que es mostra a la figura (3.). Aquest cop havia d’anar amb molta més cura per tal que el disseny es mantingués homogeni, però no fos una còpia, ja que cada pàgina és pertinent que tingui característiques concretes que la distingeixin de les altres. Per exemple, en aquest cas, la creu per afegir llistes de l’anterior apartat va passar a ser el símbol de compartir. També vaig afegir un quadre de text per poder afegir els productes que anirien a la llista de l’esdeveniment. Aquesta forma és més discreta i indica de manera indirecta a l’usuari que és una funció de “menor importància”, és a dir, que no és un esdeveniment, sinó un producte del mateix.

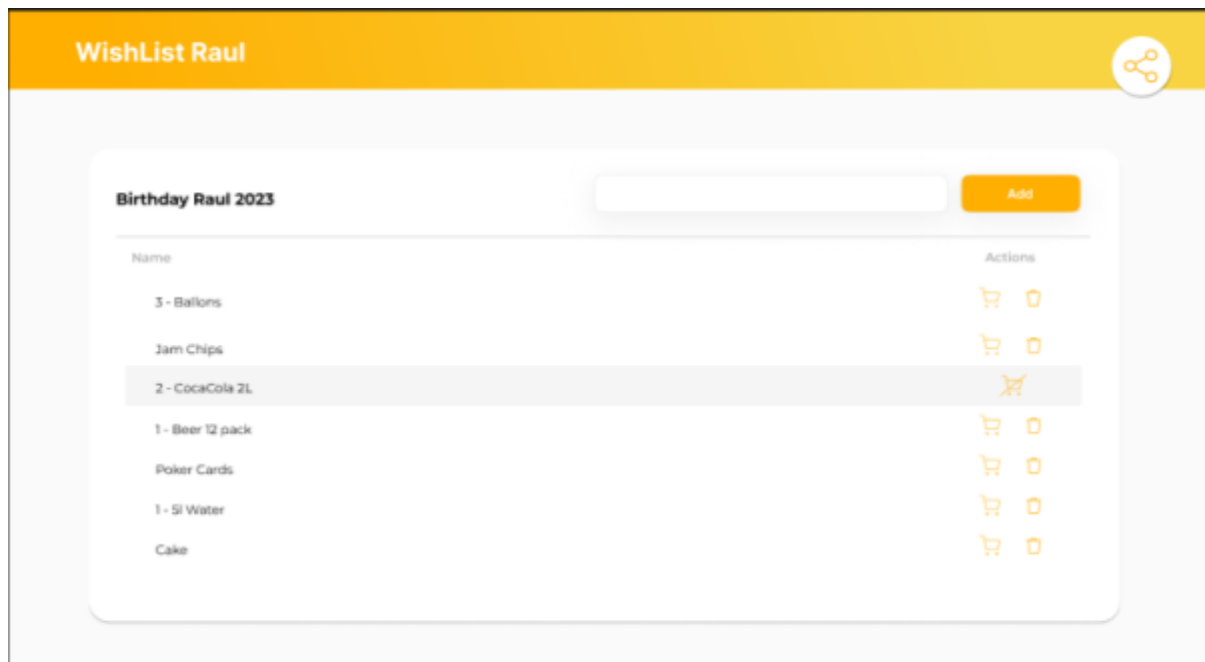


Figura 3.12 Esdeveniment amb productes

Per altra banda, hi tenim el carro, que servirà per indicar que el producte ha estat adquirit. El podrem marcar i desmarcar tants cops com vulguem. Un cop seleccionat no es podrà esborrar l'element de l'esdeveniment, ja que no tindria lògica borrar un producte que algú ha adquirit.

Contigu al carro hi trobem la paperera, la qual ens permetrà esborrar els elements. Si la llista no ens pertany no podrem esborrar els seus elements, però sí afegir-los i marcar-los com comprats.

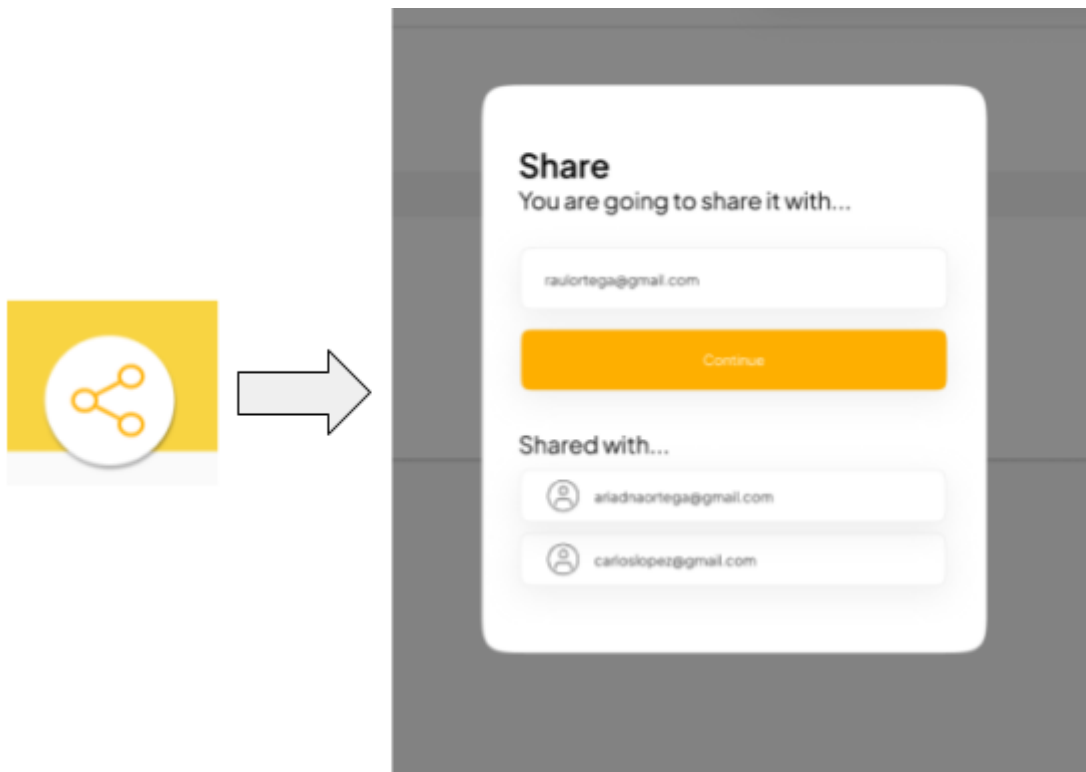


Figura 3.13 Acció en clicar el botó de compartir

Finalment, un cop cliquem sobre el botó de compartir, com es mostra a la imatge () s'obre un "modal" que ens permet compartir l'esdeveniment amb qui vulguem i veure amb qui l'hem compartit prèviament.

Quan vaig tenir el disseny acabat vaig decidir que una opció molt interessant seria la de poder escollir entre diferents modes de color. I vaig programar dos més. El que he utilitzat per a la presentació és el mode clar, i vaig crear l'obscur i el blau els quals podem observar a les figures () i (). En cas que el navegador estigui en mode clar o obscur la web ho detectarà i seleccionarà el mode corresponent.

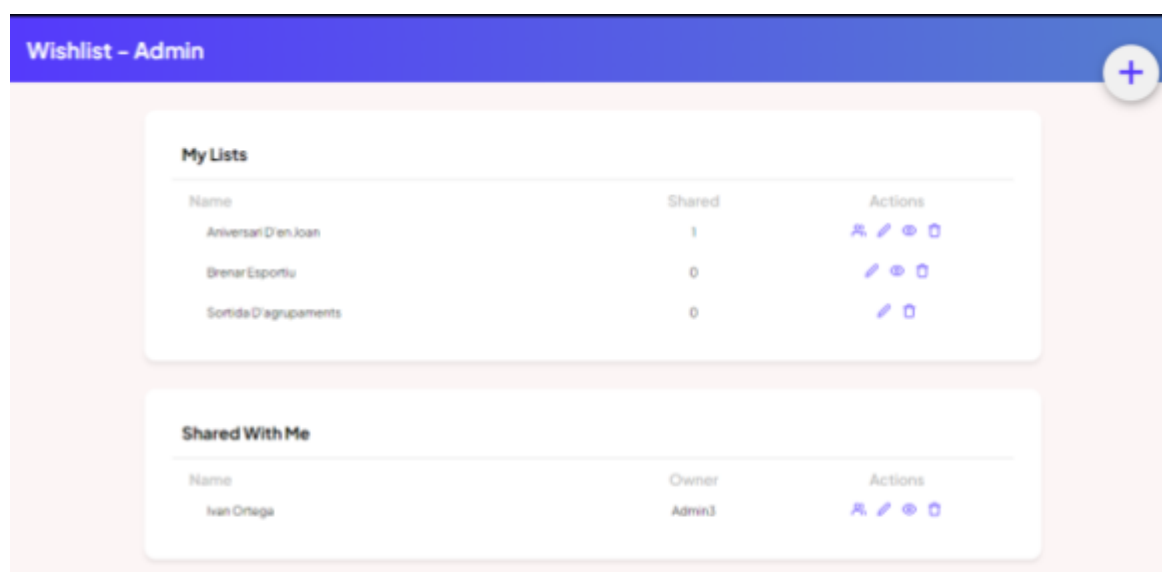


Figura 3.6.2. Disseny de la pàgina principal de la web (en blau)

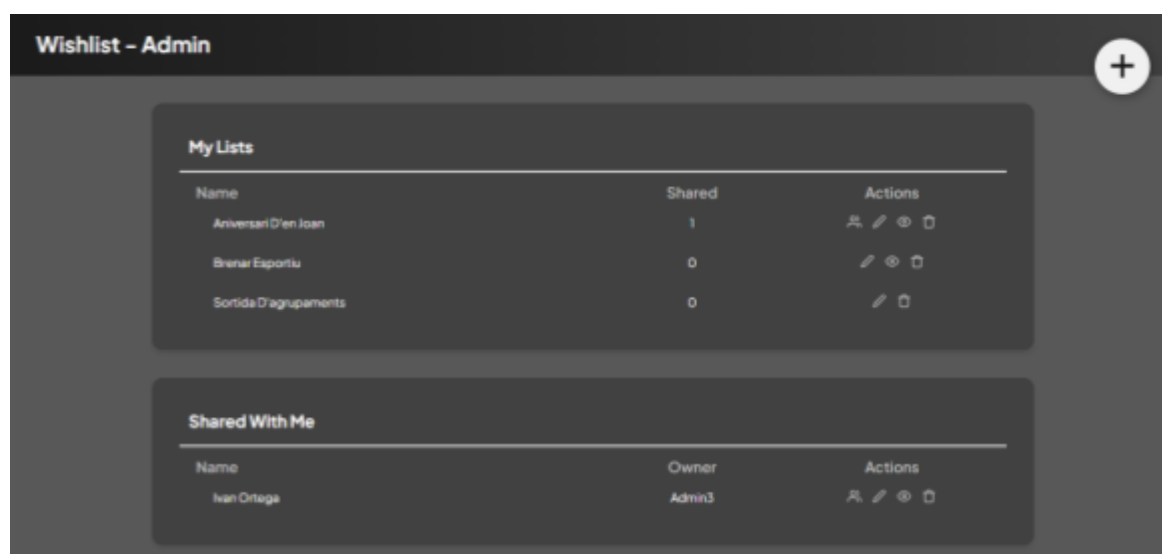


Figura 3.6.3. Disseny de la pàgina principal de la web (En mode fosc)

3.3. HTML5

El HTML5 (“Hyper Text Markup Languages”, versió 5) o bé (Llenguatge de Marcat d’hipertext), és un llenguatge descriptiu que especifica l’estructura de les pàgines web. No pot ser considerat un llenguatge de programació perquè el seu comès no és la funcionalitat de la web, sinó l’estructura. La versió definitiva va ser publicada l’octubre del 2014 i actualment és l’únic programa identificat pels navegadors capaç de fer les funcions que ara esmentarem. Funciona per etiquetes, que és el nom que reben totes les entrades amb què creem el codi (<header>, <div>, <>...) Aquest crea la base del nostre codi (textos, qüestionaris, requadres...), i estructura la part que visual a la nostra pàgina web i també l’organitza. Per tal de fer-ho correctament hi ha dos patrons d’organització per seguir.

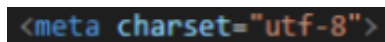
Primerament, a l’hora de crear una pàgina web utilitària hi ha diverses parts, tals com l’inici de sessió (“login”), registre (“register”), índex... Cadascuna d’aquestes parts té una funció diferent i, per tant, per tenir-ho tot ordenat s’han de crear diversos fitxers d’HTML5, on en cadascun d’ells hi haurà el que volem mostrar a cada pàgina. Això es deu al fet que és gairebé impossible posar-ho tot dins d’un mateix fitxer, ja que es mostraria tot el contingut en una mateixa pantalla. Per aquest motiu creem distincions entre les diferents parts de la pàgina, per mostrar el contingut de forma clara i raonable. Un cop hem creat els diferents fitxers d’HTML els haurem d’enllaçar entre ells, i, tenint en compte l’acció de l’usuari redirigir-lo a l’un o l’altre.

L’inici de sessió i el registre, com indiquen els seus noms ens serviran per crear les nostres credencials a la pàgina i poder entrar-hi. A l’índex, en canvi, és on anirà tot el contingut de la pàgina web. Aquest fitxer el podem anomenar com vulguem, però índex és la forma estàndard i és el primer arxiu que busca el cercador al nostre codi. Per tant, és recomanable utilitzar aquest nom per optimitzar l’experiència d’usuari.

En segon lloc, trobem l’organització a l’interior de cada fitxer HTML5, que té una estructura que hem de seguir. Aquesta estructura té dues parts principals, i, depenent del projecte més o menys subapartats.

La primera part d'aquesta estructura és el "head", el qual emmagatzema i proporciona informació de metadades, les quals proporcionen informació al cercador sobre els paràmetres de la web; com la codificació, les dimensions, l'escala...

1. Inicialment, tenim el "meta charset", figura (3.13), el qual indica quin tipus de codificació tindrà la pàgina i el qual ha d'anar en primera posició després del "head". Així serà el primer que trobarà el cercador web, reconeixerà aquesta codificació i llegirà el codi correctament. No és completament obligatori, ja que, per defecte HTML5 serà interpretat amb aquest "meta charset". Però si l'especifiquem cercadors mal configurats no tindran la possibilitat de mostrar incorrectament el nostre projecte. Actualment, la forma de codificació més emprada es anomenada "utf-8".



```
<meta charset="utf-8">
```

Figura 3.13 Metadada

2. En segon lloc, trobem una altra "meta dada" que indica l'amplada i l'escala de la pàgina, figura (3.14). Això és molt important, pel fet que definirà com veiem la pàgina web, i si es pot fer zoom de forma correcta o no. El més habitual és fer ús de la instrucció que es pot veure a la imatge 3.4. Aquesta configuració adapta el contingut de la nostra pàgina web a l'amplada real del dispositiu i manté les proporcions. Això és possible perquè li indiquem que l'amplada ha de ser la del dispositiu i que ha de mantenir l'escala inicial a 1. Com que només definim l'escala inicial, l'usuari podrà augmentar la web sense cap mena de problema, permetent així que la web sigui adaptativa.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Figura 3.14 Metadada

3. La penúltima dada que trobarem al “head” es compon de dues formes d’enllaç (<script>, figura (3.15), i <link>, figura (3.16)). El “script” que trobarem al “head” serà el qual ens indicarà la versió de “jquery” que trobarem (que com explicarem més endavant és una eina que ens ajudarà a enllaçar la web amb els servidors de la xarxa en línia). En canvi, el “link”, com indica el seu nom, serà un vincle amb el CSS+, que permetrà que els estils que definim al CSS s’apliquin a l’HTML5.

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

Figura 3.15 Metadada

```
<link rel="stylesheet" href="/style.css">
```

Figura 3.16 Metadada

4. Finalment, trobarem el títol, que sortirà a la pestanya de la pàgina, figura (3.17).

```
<title>API Wishlist</title>
```

Figura 3.17 Metadada

Un cop hem vist perquè serveixen aquestes etiquetes, aquí podem observar, a forma d'exemple, l'inici de la meua pàgina web, figura (3.18). Hi torbem definides

totes les dades mencionades i podem observar l'estructura. Cal recalcar que no és necessari seguir un ordre concret amb aquestes etiquetes, però és usual trobar els codis en l'àmbit professional d'aquesta manera. Podríem dir que se segueix un conveni no estipulat de forma oficial, per fer-ho tot més entenedor i per ajudar als motors de cerca (els navegadors) a tenir la major eficiència i rapidesa.

```
<!DOCTYPE html>
<html>

<head>
  <title>API Wishlist</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <link rel="stylesheet" href="/style.css">
</head>
```

*Figura 3.18 Iníci del HTML de la web
(conjunt de metadades)*

La segona part d'aquesta estructura és el "body". Aquesta contindrà tot el codi que es mostrarà en pantalla i, com a la resta de casos, és recomanable que segueixi un ordre. Si ho fem correctament el cercador web ho interpretarà més ràpidament i l'experiència d'usuari serà més agradable i fluida.

1. En primer lloc, trobarem el "header", totalment diferent del "head", el qual ja hem explicat. Aquesta etiqueta contindrà tot allò que aparegui a la capçalera de la nostra pàgina.
2. Seguidament tenim el cos. En aquest cas no té una etiqueta específica, ja que, ja som dins del "body", per tant, no cal distingir-lo. Aquí escriurem tot el que sigui el cos principal de la pàgina. Però aquest també haurà de seguir una estructura, que principalment es basarà en els (<div>), els quals són una mena de contenidors d'informació.

2.1. El primer mètode d'organització és el ("class"), figura (3.19). Les classes són distintius que ens serviran més endavant per afegir funcionalitat o estil als elements. Poden estar associades a un o diversos elements, això ens facilitarà el treball, perquè, com ja hem mencionat a l'apartat del disseny, una pàgina web ha de seguir una mateixa línia de disseny en tots els seus apartats. Per tant, hi ha parts que han de tenir el mateix estil base (per exemple dues llistes com les de l'usuari i les compartides), on puc aprofitar tot l'estil exterior i només he de canviar els botons d'acció i el contingut que es mostra en elles.

```
<div class="body-header">
```

*Figura 3.19 classe, forma d'organització
a l'HTML*

2.2. La segona manera d'organitzar el contingut és el que es coneix com a "id", figura (3.20). Aquest atribut és un identificador, tal com el DNI, que és únic per a cada element. Ens serà molt útil, tant per estilitzar elements concrets, com per afegir funcionalitat a la pàgina web. Profunditzarem en com fer-ho més endavant (quan expliquem JavaScript), però bàsicament gràcies al "id" podrem afegir i eliminar elements varis de forma individual.

```
<button id="myBtn">+</button>
```

Figura 3.20 Id, forma d'organització

Un cop coneixem aquestes dues formes de donar ordre a la pàgina web, cal mencionar també que no són exclusius pel div. Tot i que el (<div>) és el principal element per separar i emmagatzemar contingut en el nostre codi HTML, aquests distintius també serveixen per a altres etiquetes, sobretot el "id". L'identificador és clau a l'hora de distingir elements, des de botons, fins a quadres de text o elements d'una llista. I les classes, com hem dit serviran

sobretot per a afegir estils tot i que també ens ajudarà a fer funcionar algunes parts de la pàgina web

També cal mencionar que en aquest apartat es fa un (`<script>`), el qual ens enllaça l'HTML amb el JavaScript, fet molt rellevant perquè com veurem és essencial en la creació de pàgines web utilitàries.

3. Per acabar trobem el “footer” o peu de pàgina. En aquest apartat podem posar-hi la informació que trobem necessària. Tot i que el més recomanable és posar-hi enllaços a parts d'interès de la web, parts importants o bé diferents formes de contacte (correu electrònic) o xarxes socials (Instagram, Twitter).

3.4. CSS+

El CSS+ o bé (“Cascading Style Sheets”— Fulles d'estil en cascada) és l'eina que donarà l'estil a la nostra pàgina web. Serveix per crear un entorn accessible i entenedor, ja que, una pàgina sense l'estil és complicada d'entendre i utilitzar, a més de poc visual.

El CSS ens permetrà compartir un sol estil a diversos documents HTML i tindrem una gran flexibilitat a l'hora de donar configuracions estètiques a un element particular. Podrem realitzar aquestes accions per mitjà d'etiquetes que ens permetran canviar des de la mida de la font fins a l'amplada dels contenidors d'informació. En aquest cas tampoc és considerat un llenguatge de programació pel fet que la seva funció és l'estètica del contingut i no la funcionalitat d'aquest.

Aquest llenguatge està estrictament relacionat amb l'HTML, directament el modifica. Per fer-ho pertinentment segueix la seva pròpia estructura jeràrquica.

Com ja hem explicat a l'hora que creem una pàgina web és molt important que l'estètica estigui cuidada, però, més encara que sigui concordant amb si mateixa. Generalment, no podem utilitzar una gamma cromàtica molt àmplia, ni un format diferent per a cada apartat. Per tant, com que molts apartats coincidiran, podem reduir enormement la feina fent un sol full d'estils, on canviarem l'estil de tota la web.

I llavors, mitjançant l'etiqueta ("class") o bé el nom de qualsevol altre etiqueta (si modifiquem l'etiqueta <button> tots els botons rebran la mateixa modificació) per només haver de definir un sol cop l'estil, i no cada cop que ens trobem un mateix element haver de repetir el procés.

Per exemple si volem que tots els enllaços siguin d'una forma concreta podríem definir al CSS que l'etiqueta ("<a>") serà d'una manera i tots els HTML que tinguin enllaçat aquell full d'estils aplicaran la mateixa estètica a tots els seus botons. Figura (3.21)

```
a{  
  text-decoration: none;  
  color: var(--primary);  
  font-weight: bold;
```

Figura 3.21 Atribució a l'etiqueta <a> d'uns estils determinats

Alguns cops, però, no voldrem que tots els botons siguin iguals. Voldrem que els d'un apartat tinguin un color i els d'un altre un de diferent. Llavors entrarà en joc l'etiqueta ("<class>"), que permet afegir als botons una classe i nosaltres modificarem aquella classe cridant-la amb un punt davant del nom, figura (3.22). Com veiem a la figura() es modificaran els botons que estiguin dins de la classe ("list-adder")

```
.list-adder button{
  border-radius: 10px;
  border: 1px solid var(--background);
  background: var(--primary);
  height: 32px;
  width: auto;
  padding: 10px;
  color: var(--buttonColor);
  margin-left: 10px;
  cursor: pointer;
}
```

Figura 3.21 ús del classe per afegir estils

També, pot ser que vulguem modificar un o més elements que es troben en un espai concret, figura (3.22). Per fer-ho utilitzarem la jerarquia del codi i farem un llistat dels elements pels quals hem de passar per poder arribar al que volem. En altres paraules si volem editar un el cos d'una taula que es troba en dins una taula, que al seu torn es troba a un de la classe (".my-lists"); primer cridarem a la classe, després a la taula i finalment al cos de la taula.

```
.my-lists table tbody{
  padding-left: 40px;
  text-align: center;
}
```

*Figura 3.22 Modificació
d'estils per mitjà de la
jerarquia de CSS*

Finalment, potser el que volem és modificar un element concret, sense afectar a cap altre. Per fer-ho farem servir l'id. Aquest és únic per l'etiqueta que creem i no l'haurem de repetir. La manera de cridar aquesta funció és fer servir un coixinet tot just abans de l'id de la nostra funció. L'id l'haurem d'haver definit nosaltres mateixos prèviament.

```
#purchased{  
  opacity: 50%;  
}
```

*Figura 3.23 Ús de l'id
per afegir estils*

Cal incidir en com està organitzada l'estructura de prioritat del CSS. Té dues condicions que fan més o menys prioritàries les modificacions a l'estil que programem.

Primerament hi ha una prioritat per especificitat. Com més específics siguem més rellevància té allò que escrivim sobre la funció. Un (“<id>”) tindria prioritat sobre un (“<class>”) pel fet que és d'un element específic. També ho podem veure reflectit en casos com hem vist anteriorment, on volem modificar un element i hi accedim a través de cridar altres funcions, com més en cridem, més concret i conseqüentment més importància tindrà aquest element. Per exemple, en el cas de la figura (3.24) la segona funció tindria prioritat davant la primera. La primera crida la classe general “modal-buttons”. En canvi, la segona crida a la mateixa classe, però a més a més al botó, per tant, és més específica i si hi hagués estils no concordants es farien servir els de la segona cadena.

```
.modal-buttons{
  margin-top: 40px;
  display: flex;
  justify-content: space-between;
}

.modal-buttons button{
  border-radius: 10px;
  border: 1px solid var(--background);
  background: var(--primary);
  height: 52px;
  width: 45%;
  padding: 10px;
  color: var(--buttonColor);
  cursor: pointer;
}
```

Figura 3.24 Prioritat per especificitat

En segon lloc, trobem, com indica el nom, una prioritat de salt d'aigua. És a dir, com més avall en el codi sigui una funció més importància té. Això es deu al fet que CSS interpreta el codi de dalt a baix i, com a resultat, es queda amb l'últim valor de la funció. Aquí caldrà tenir en compte que aquesta prioritat només s'aplicarà a funcions d'un mateix nivell. En cas que no tinguin el mateix nivell es respectarà la primera jerarquia mencionada. En l'exemple de la figura (3.25), tot i que no té lògica, trobem dues funcions iguals, que tenen la mateixa especificitat. Per tant, el que farà el codi és interpretar l'últim valor d'aquesta funció, és a dir, aquesta funció tindrà un ("<margin-bottom>") de 20 píxels i no de 40 píxels.

```
.modal-content h2{
  margin-bottom: 40px;
}

.modal-content h2{
  margin-bottom: 20px;
}
```

Figura 3.25 Prioritat en salt d'aigua

En el meu cas a l'hora de realitzar aquest apartat de la pàgina web el primer que vaig haver de fer, va ser pensar que havia de contenir. La majoria dels estils (login, register, index...) havien de tenir un aspecte similar, i a més, pertànyer al mateix CSS. Per tant, un cop enllaçat el full d'estils vaig iniciar la programació. El primer que vaig afegir fou la definició dels colors principals de la pàgina, figura (3.26) els quals s'han de fer servir molt sovint al llarg del codi i per raons estètiques han de ser sempre iguals. A més ens permetrà canviar els colors de tota la pàgina d'una fora molt més pràctica.

```
:root {  
  --white: #fff;  
  --primary: #FEAF00;  
  --secondary: #F8D442;  
  --buttonColor: white;  
  --gray: #bebebe;;  
  --text: black;  
  --icons: var(--primary);  
  --background: rgb(252, 245, 245);  
  --box-background: white;  
}
```

Figura 3.26 Definició dels colors a CSS

Tot seguit vaig definir la font, figura (3.27). En el meu cas particular la vaig exportar, pel fet que, en volia una de concreta i no es trobava entre les predefinides que té HTML. Per fer-ho vaig haver de buscar la font i el format que moltes pàgines ja t'entreguen junts i definir el ("font-weight"), que és el pes o amplada de la lletra.

```
@font-face {  
  font-family: "customFont";  
  src: url("/fonts/PlusJakartaSans-VariableFont_wght.woff2") format("woff2 supports variations"),  
       url("/fonts/PlusJakartaSans-VariableFont_wght.woff2") format("woff2-variations");  
  font-weight: 100 1000;  
}
```

Figura 3.28 Declaració de la font a CSS

Finalment, vaig iniciar amb la descripció estilogràfica de la resta d'elements de la meua pàgina, fent ús dels termes comentats.

3.5. JavaScript

El JavaScript és el component més difícil d'entendre i utilitzar a una pàgina web, però també el més important. Amb el JavaScript podem afegir la funcionalitat a la pàgina (des de crear els elements d'una llista fins a fer que el botó que cliquem realitzi una acció) i també permet connectar la pàgina web, mitjançant apis al servidor. Si no duguéssim a terme aquesta connexió no podríem visualitzar el contingut multimèdia a diferents dispositius, quedaria reduït a l'àmbit local, que tot i que aquest també és útil per si sol no compliria amb el que requereix una pàgina web. Aquest sí és considerat un llenguatge de programació, perquè sí que afegeix la funcionalitat a la pàgina.

A l'hora de fer servir JavaScript hi ha diferents opcions. Depenent de l'opció que fem servir canviarà una mica la forma de treballar o donarà funcionalitats que en un altre JavaScript més bàsic no hi són. Al meu treball he utilitzat dues d'aquestes formes: el jquery i l'ajax.

El jquery és una biblioteca de JavaScript que permet simplificar la forma d'interactuar amb els documents HTML i permet la implementació d'interacció amb la tècnica ajax per a les pàgines web. Aquesta simplificació del codi a l'hora de programar és altament eficaç, permet estalviar-se moltes línies i arribar a la mateixa funció. A més permet aprendre termes més curts i fàcils de recordar i minimitzar els

errors, He utilitzat aquesta biblioteca perquè simplifica molt la programació i té unes funcionalitats molt bones i molt treballades.

L'ajax, en canvi, és una tècnica de desenvolupament de pàgines web asíncrones. Això vol dir que mentre nosaltres interactuem amb la pàgina web aquesta continua carregant el codi mentre realitza la funció esperada. Quan nosaltres volem entrar en un esdeveniment o alguna llista el que fem és clicar un botó. Un cop ajax ho detecta envia aquesta informació a les api i el servidor espera a rebre la resposta que ha de mostrar en pantalla. Mentre du a terme aquesta funció el codi no es queda estancat en aquella zona del botó, sinó que continua carregant la pàgina web i quan rep la informació de l'acció realitzada la mostra en pantalla. Això permet que les pàgines web siguin molt més fluides i utilitzables, pel fet que si haguéssim d'esperar cada cop que volem realitzar una acció la web no acabaria mai de carregar i no seria útil.

Una altra de les funcions de JavaScript és la capacitat d'accedir a l'emmagatzematge local del dispositiu ("local storage") Aquest ens serà molt útil per dur a terme accions com mantenir el token de l'usuari i mantenir l'inici de sessió obert.

L'autenticació basada en tokens és una forma de verificar la identitat mitjançant els tokens. Els servidors els fan servir per verificar la identitat d'un usuari, una api, un ordinador o fins i tot un altre servidor. Els tokens són elements simbòlics i temporals que expedeix una font de confiança, és com una placa distintiva, i n'hi ha de físics i digitals.

Els físics (contrasenya) són demanats durant l'inici de sessió per verificar la identitat de la persona i poder accedir allà on sigui necessari.

Els tokens web, en canvi, són generats per un servidor i enviats a l'usuari quan s'ha realitzat un inici de sessió exitós. Aquests permeten saber al servidor qui és l'usuari i a quina informació té accés.

Tot aquest procés ens permetrà no haver d'iniciar sessió sempre, ja que, el servidor identificarà el token que ell mateix ens va compartir i ens permetrà accedir a la informació a la qual tinguem permís.

Més enllà de la connexió amb servidors el JavaScript és el motor de les webs. Afegeix totes les funcionalitats i accions que aquestes pàgines puguin dur a terme.

En primer lloc, hem de crear una variable a la qual afegirem la funcionalitat que sigui necessària. Per fer-ho trobarem tres maneres diferents de definir les variables.

La primera s'anomena ("var"): Aquesta variable és la més general de totes. Té un abast de funció, que significa que quan definim una variable d'aquesta manera existirà a tota la funció que definim. Aquesta pot ser reemplaçada, per tant, el valor inicial no és fix. Actualment, es troba en desús, pel fet que,

En segon lloc, trobem ("const"): La variable const a diferència de la primera té un abast de bloc. Això abasta des de l'obertura dels claudàtors fins al tancament d'aquests. Això vol dir que la variable només existirà entre els claudàtors que l'haguem creat, i, si la cridem fora d'aquest bloc ens saltarà un error, perquè la variable no estarà definida, no existirà. En aquest cas el valor de la variable sí que és fix. Un cop el definim si provem de reassignar-li un valor ens donarà un error, perquè és un valor constant.

Finalment, trobem la variable ("let"): És una fusió de les dues anteriors. És una variable amb abast de bloc. Per tant, com la variable ("const"), només existirà dins del bloc on sigui creada. Però aquesta no serà de valor fix. Podrem reassignar-li tants valors com vulguem que aquests se sobreescriuran a l'anterior, tal com passava amb la variable ("var").

A l'hora de programar he fet servir les dues últimes, depenent del cas una era més adient que l'altre. He prescindit de la primera, perquè com ja he mencionat té un abast massa general. Això impedeix que variables que han de ser iguals puguin ser-ho perquè els valors d'un bloc influeixen en tots els altres. A més tot i no tenir aquest problema com ja he mencionat perquè un codi sigui correcte se sol seguir un

conveni no especificat, i la millor forma de definir variables, com ja hem mencionat, és amb un abast de bloc.

A la figura (3.28) podem observar que hi ha declarades diverses funcions “const”. Han de ser constants pel fet que les funcions a les quals estan associades no canviaran mai el seu valor. En canvi, com podem veure a la figura (3.29), just després de definir la variable ja canviem el seu valor, per tant, no pot ser constant.

```
getWithAuthorization('/wishlists/' + elem.id + '/products')
  .done(function (response) {

    const modalView = document.getElementById("viewModal");
    const viewModalTitle = document.getElementById("viewModalTitle");
    const viewModallist = document.getElementById("viewModallist");
    viewModalTitle.innerText = elem.obj['name'];
    modalView.style.display = "block";
```

Figura 3.28 Ús de la variable const

```
let url = "/edit/?id=" + elem['id'];
if(shared){
  url += "&shared=true";
}
window.location.href = url;
});
```

Figura 3.29 Ús de la variable let

Un cop sabem com crear les variables, ja en podem fer ús d'elles i programar el nostre JavaScript sense problemes.

Per afegir funcionalitat als botons hi trobem múltiples opcions. Totes tenen la mateixa finalitat, però funcionen de manera diferent. Haurem d'escollir el que més s'adapti a nosaltres i a cada situació.

Primerament torbarem el ("addEventListener"): Aquest ens permet afegir múltiples funcionalitats a una variable (normalment un botó). Podem afegir tantes accions com vulguem que totes s'executaran al mateix temps a l'hora de prémer el botó. Per fer-ho simplement afegim diverses vegades l'etiqueta i indiquem la funció que volem que faci. A l'hora d'indicar la funció que volem que realitzi, també haurem d'indicar quina acció cal fer perquè allò passi (per exemple; en clicar). És una funció exclusiva de JavaScript, per tant, només la podrem fer servir en el mateix, i no a l'HTML. Aquesta etiqueta és bastant nova i no la suporten els navegadors més vells. No obstant qualsevol ordinador mitjanament actual pot actualitzar el seu navegador a l'última versió sense cap problema.

En segon lloc, hi ha l'etiqueta ("onclick"): Aquesta ens permetrà afegir una sola funcionalitat al botó en qüestió. A diferència de l'altre etiqueta si afegim una segona funció aquesta se sobreescriurà a la primera i només serà executada l'última. En aquest cas només haurem d'indicar quina funció es farà en clicar el botó. Aquesta etiqueta no és exclusiva de JavaScript, també en podem fer ús a l'HTML. El ("onclick") és una funció «antiga» i en conseqüència és compatible amb tots els navegadors.

A la meva pàgina hi podem trobar les dues etiquetes, tot i que, hi predomina ("AddEventListener"), perquè és més afí amb les funcionalitats que he de realitzar a la meua pàgina web.

4. Conclusions i possibles millores.

Un cop acabat el treball puc dir que he aconseguit la majoria dels resultats pels quals vaig iniciar aquest projecte. He après moltes coses sobre el disseny de pàgines web i a programar en HTML, CSS i JavaScript partint des de zero. M'he familiaritzat prou amb aquests llenguatges de programació per a poder programar pàgines web d'un cert nivell de complexitat. També sobre apis i servidors per poder connectar tot allò que programi a la xarxa. Tot això m'ha permès programar la meua pròpia pàgina web des de zero. Aquesta és tal com la vaig dissenyar, fins i tot amb algunes millores al disseny inicial (opcions de color variades, millora de les funcionalitats...).

Tot i això, penso que hi ha diferents aspectes a millorar. I com que la meua pàgina web és un projecte personal tinc la intenció que amb el temps pugui anar implementant aquestes millores

Possibles millores:

- Afegir diferents rangs per tal que alguns usuaris no puguin modificar les llistes i altres sí.
- Possibilitat d'introduir la quantitat gastada a cada producte i el càlcul de què deu cada persona a un altre.
- Inici de sessió ràpid mitjançant Google o Apple.

Després de tot el treball he pogut extreure diverses conclusions interessants. Primerament, la programació és molt més complexa del que pot semblar en un inici. Has d'aprendre diversos idiomes i a més aprendre a controlar sistemes de connexió amb la xarxa i els servidors, jo he necessitat ajuda externa en molts casos. En segon lloc, he pogut comprovar que la constància és una eina molt important a l'hora de realitzar qualsevol feina. Com més ens relaxem en un àmbit més ens costarà i encara tindrem menys ganes de tornar-nos-hi a posar. Finalment, també he pogut comprovar la versatilitat de la programació. Si sabem programar podrem

realitzar infinitat de projectes, com ara la web que jo he realitzat o bé, un petit menú als annexos on es pot accedir a diferents pàgines d'interès o descarregar el treball.

5. Agraïments

En primer lloc, vull agrair a l'Antònia Subirà Ayats, la meva tutora i directora de l'institut Jaume Callis, haver-me guiat en tot el procés. Tant en el descobriment i elaboració de la pàgina web com en la redacció del treball escrit. Penso que ha estat una sort poder tenir una tutora tan implicada en el meu treball. Arribant a fer reunions setmanals durant pràcticament tot el projecte. I també procurant que em mantingués sempre centrat i no deixes de treballar. Considero que gràcies a la seva flexibilitat sense descuidar-me he pogut gaudir més de fer aquest treball. Per tot això penso que no hagués pogut tenir millor tutora.

Per altra banda, també vull donar les gràcies a Ivan Ortega Asensio, expert programador, per haver-me donat suport a allò que l'Antònia menys podia, la programació. Gràcies a les seves pautes i ajudes he pogut realitzar aquest apartat del treball amb una dificultat menor. Permetent-me assolir la majoria dels meus objectius finals.

6. Fonts d'informació

27 de juny del 2023: https://youtu.be/Y-OhzQpsRwI?si=V_niYAhP0nOlX05 Curs Bàsic d'HTML5 i CSS3 des de zero - devichol

27 de juny del 2023: <https://youtu.be/OuoLQiXPitk?si=Tm8wx8Yc1WrwO-BV> Cómo Hacer un Encabezado Responsive con HTML y CSS - deivichol

12 de juliol del 2023: <https://youtu.be/jk1T0CdLxwU?si=8vklXyyiadpis8YD> Intro to Figma - Beginners guide to Figma Basics - Jesse Showalter

7. Annexos

7.1. Accés a un menú amb la versió digital del projecte i la pàgina web. Figura (7.1)



Figura 7.1 Codi qr al menú del meu treball

7.2. Accés a una carpeta de drive amb totes les pràctiques de programació realitzades. Figura (7.2)



Figura 7.1 Codi qr al menú del meu treball