

La Maria WebApp

Desenvolupament d'una aplicació web per a la promoció de La Maria

The logo of the Universitat Oberta de Catalunya (UOC), consisting of the letters 'UOC' in a bold, blue, sans-serif font.

Roser Fuster Bertomeu

Màster universitari Online de
Desenvolupament de Llocs i
Aplicacions Web

Informàtica, Multimèdia i
Telecomunicació

Tutora de TF

Anna Ferry Mestres

**Professor responsable de
l'assignatura**

César Pablo Córcoles Briongos

Universitat Oberta
de Catalunya

16 de juny de 2024

Llicència

© Roser Fuster Bertomeu

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual

Agraïments

Primerament, agrair a la meua tutora, Anna, els consells que m'ha donat tan útils, la seua experiència i disponibilitat a l'hora de resoldre dubtes que han estat tan presents a l'inici d'aquest treball.

Agrair, també, a la meua família per tota la paciència, el suport constant i estima incondicional que m'heu brindat durant els dos anys d'aquest màster i sobretot en aquest projecte. Heu sigut pilars fonamentals per a seguir endavant.

Probablement, la persona que més ha patit tots els alts i baixos d'aquesta aventura és la meua parella. Sense les seues paraules d'ànim sempre oportunes, no hauria superat aquests dos anys tan durs. Gràcies per la teua infinita estima.

I, per descomptat, gràcies a La Maria, sense ella este projecte mancaria de sentit.

A totes i tots, moltes gràcies. Aquest treball és tant vostre com meu.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>La Maria WebApp</i>
Nom de l'autor:	<i>Roser Fuster Bertomeu</i>
Nom del consultor/a:	<i>Anna Ferry Mestres</i>
Nom del PRA:	<i>César Pablo Córcoles Briongos</i>
Data de lliurament (mm/aaaa):	<i>06/2024</i>
Titulació o programa:	<i>Màster de Màster universitari Online de Desenvolupament de Llocs i Aplicacions Web</i>
Àrea del Treball Final:	<i>Informàtica, Multimèdia i Telecomunicació</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Angular, Laravel, música</i>
Resum del Treball	
<p>En aquest projecte es documenta el desenvolupament d'una aplicació web que conté una plataforma web i el seu gestor de contingut per a l'artista musical La Maria, una música i cantant emergent en el panorama musical valencià. Aquesta aplicació web mostrarà públicament la informació relacionada amb l'artista. El gestor de contingut permetrà a qui gestiona dita informació, administrar-la.</p> <p>Aquest projecte està dissenyat amb una arquitectura de tres capes. La capa client està desenvolupada amb Angular, l'API per a la gestió de les dades està desenvolupada amb Laravel, i com a motor de dades s'utilitza MySQL.</p> <p>Tant el portal web com el gestor de contingut estan dissenyats de manera adaptable per a poder fer-ne ús des de tots els dispositius.</p> <p>El producte final, ofereix una solució tecnològica accessible i actualitzada per a promocionar en el món digital a La Maria.</p>	
Abstract	
<p>This project documents the development of a web application containing a web platform and its content management system for the emerging musical artist La Maria, known in the Valencian music scene. This web application will publicly display information related to the artist. The content manager will allow those who manage this information to administer it.</p>	

The project is designed with a three-tier architecture. The client layer is developed using Angular, the data management API is developed with Laravel, and MySQL is used as the data engine.

Both the web portal and the content manager are designed to be adaptable for use on all devices.

The final product offers an accessible and up-to-date technological solution for promoting La Maria in the digital world.

Índex

1.	Introducció.....	1
1.1.	Context i justificació del Treball.....	1
1.2.	Objectius del Treball.....	2
1.3.	Impacte en sostenibilitat, ètic-social i de diversitat.....	2
1.4.	Enfocament i mètode seguit.....	3
1.5.	Planificació del Treball.....	5
1.6.	Breu sumari de productes obtinguts.....	5
1.7.	Breu descripció dels altres capítols de la memòria.....	6
2.	Materials i mètodes.....	7
2.1.	Anàlisi de requisits.....	7
2.2.	Arquitectura de l'aplicació.....	9
2.3.	Disseny de l'aplicació.....	12
2.4.	Prototipatge.....	24
3.	Desenvolupament.....	36
3.1.	Back-End.....	36
3.2.	Front-End.....	39
4.	Tests.....	44
4.1.	<i>Back-End</i>	44
4.2.	<i>Front-End</i>	45
5.	Desplegament de l'aplicació.....	46
5.1.	Requeriments.....	46
5.2.	Desplegament <i>back-end</i> en local.....	46
5.3.	Desplegament <i>back-end</i> en el servidor.....	47
5.4.	Desplegament <i>front-end</i> en local.....	49
5.5.	Desplegament <i>front-end</i> en el servidor.....	50
6.	Instruccions d'ús.....	52
6.1.	Portal públic.....	52
6.2.	Portal privat.....	57
7.	Bugs.....	65
8.	Conclusions i treballs futurs.....	66
9.	Glossari.....	68
10.	Bibliografia.....	69
	Annex 1. Extractes de codi font <i>back-end</i>	71
	Annex 2. Extractes de codi font <i>front-end</i>	73

Llista de figures

Figura 1: La Maria amb el disc L'Assumpció [1].	1
Figura 2: Metodologia de desenvolupament en cascada	4
Figura 3: Diagrama de Gantt	5
Figura 4: Diagrama d'arquitectura del sistema	10
Figura 5: Diagrama de casos d'ús	13
Figura 6: Sitemap de l'aplicació	15
Figura 7: Diagrama d'activitat sistema portal web	17
Figura 8: Diagrama d'activitat sistema d'autenticació	18
Figura 9: Diagrama d'activitat sistema comanda	19
Figura 10: Diagrama d'activitat sistema concert	20
Figura 11: Diagrama d'activitat sistema producte	21
Figura 12: Diagrama d'activitat sistema usuari	22
Figura 13: Model entitat-relació de l'aplicació	23
Figura 14: Wireframes d'autenticació, registre i recuperar contrasenya	24
Figura 15: Wireframes pàgina d'inici amb menú, la pàgina d'inici i concerts	25
Figura 16: Wireframe productes versió mòbil i escriptori	25
Figura 17: Wireframe música versió mòbil i escriptori	26
Figura 18: Wireframe pàgina detall del producte, cistella i reserva cistella	26
Figura 19: Wireframes menú versió mòbil i escriptori	27
Figura 20: Wireframes gestió de concerts, gestió de productes i formularis	27
Figura 21: Wireframes el meu compte i formularis	28
Figura 22: Wireframes comandes, detall comanda, gestió usuaris i formulari	28
Figura 23: Logotip de l'aplicació	28
Figura 24: Paleta de colors de l'aplicació	29
Figura 25: Comprovació de contrast del color principal	29
Figura 26: Comprovació de contrast del color secundari	29
Figura 27: Comprovació de contrast del color secundari <i>light</i>	29
Figura 28: Estil d'elements interactius de l'aplicació	30
Figura 29: Icones usades a l'aplicació	30
Figura 30: Mockups d'autenticació, registre i recuperar contrasenya	31
Figura 31: Mockups pàgina d'inici amb menú, pàgina d'inici i concerts [8][9]	31
Figura 32: Mockups productes versió mòbil i escriptori [10][11]	32
Figura 33: Mockups música versió mòbil i escriptori [12][13][14][15]	32
Figura 34: Mockups pàgina detall del producte, cistella i reserva cistella [10]	33
Figura 35: Mockups menú versió mòbil i escriptori	33
Figura 36: Mockups gestió de concerts, gestió de productes i formularis [10][11]	34
Figura 37: Mockups el meu compte i formularis	34
Figura 38: Mockups comandes, detall comanda, gestió usuaris i formulari	35
Figura 39: Codi <i>middleware</i> JWT del <i>back-end</i>	36
Figura 40: Codi <i>middleware</i> Gestor del <i>back-end</i>	37
Figura 41: Codi <i>middleware</i> Admin del <i>back-end</i>	37
Figura 42: Codi definició rutes de l'API	38
Figura 43: Rutes de l'API amb controlador i <i>middlewares</i> assignats	38
Figura 44: Funció elimina usuari del <i>back-end</i>	39
Figura 45: Configuració de rutes del <i>front-end</i>	39
Figura 46: Configuració de l'API en producció del <i>front-end</i>	40
Figura 47: Fragment del fitxer <i>ngsw-config.json</i>	40

Figura 48: Fragments de codi SCSS del <i>front-end</i>	42
Figura 49: Codi HTML del component producte-item del <i>front-end</i>	43
Figura 50: Sol·licitud realitzada en Postman	44
Figura 51: Cobertura proporcionen les proves unitàries.....	45
Figura 52: Informe dels tests unitaris amb Karma	45
Figura 53: Configuració local connexió a la base de dades	46
Figura 54: Configuració local rutes emmagatzematge imatges.....	46
Figura 55: Configuració local servidor de correus electrònics	47
Figura 56: Connexió SFTP al servidor	48
Figura 57: Configuració de producció <i>back-end</i>	48
Figura 58: Configuració connexió a la base de dades.....	48
Figura 59: Configuració servidor de correus electrònics	48
Figura 60: Configuració rutes emmagatzematge imatges	49
Figura 61: Configuració Virtual Host de l'API	49
Figura 62: Fitxer .htaccess de l'aplicació.....	50
Figura 63: Configuració Virtual Host de l'aplicació	51
Figura 64: Pàgina inicial de l'aplicació.....	52
Figura 65: Menú principal versió mòbil.....	53
Figura 66: Pàgina de concerts.....	53
Figura 67: Pàgina productes	53
Figura 68: Pàgina detall producte.....	54
Figura 69: Pàgina música.....	54
Figura 70: Pàgina cistella	55
Figura 71: Pàgina inici de sessió.....	55
Figura 72: Punt d'accés de registre	56
Figura 73: Pàgina de registre	56
Figura 74: Punt d'accés a recuperar contrasenya	56
Figura 75: Pàgina recuperar contrasenya	56
Figura 76: Menú secundari.....	57
Figura 77: Punt d'accés a la pàgina reservar	57
Figura 78: Pàgina reservar	58
Figura 79: Pàgina Les meues reserves.....	58
Figura 80: Pàgina de detall d'una comanda	59
Figura 81: Pàgina El meu compte	59
Figura 82: Formulari modificar dades personals	60
Figura 83: Formulari modificar contrasenya	60
Figura 84: Formular eliminar compte.....	60
Figura 85: Pàgina gestió de productes.....	61
Figura 86: Formulari afegir o modificar un producte.....	61
Figura 87: Pàgina gestió de concerts	62
Figura 88: Formulari afegir o modificar un concert	62
Figura 89: Pàgina gestió de comandes	63
Figura 90: Pàgina gestió d'usuaris/àries.....	63
Figura 91: Formulari modifica rol.....	64

1. Introducció



Figura 1: La Maria amb el disc L'Assumpció [1].

1.1. Context i justificació del Treball

Durant el confinament, molta gent es va dedicar a crear contingut d'entreteniment, una d'eixes persones és una familiar meua que va començar pujant cançons versionades amb l'ukelele d'acompanyament. Arran d'ací, va descobrir que volia formar-se en cant i es va matricular en el conservatori. Va ser amb una d'aquestes cançons amb les quals es va fer viral en Twitter i l'ha portada fer dues gires de concerts i a publicar un disc, sota el nom de La Maria.

Actualment, no disposa de cap portal web en el qual promocionar els seus concerts més enllà de les xarxes socials i Linktree, on pública els seus pròxims concerts i els enllaços per a comprar les entrades entre altres. Ara que ha agafat rellevància en el panorama musical valencià s'ha plantejat l'opció de tindre un portal web més professional.

Existeixen gestors de contingut com Wordpress, Wix o Blogger que ofereixen la possibilitat de generar portals web on mostrar la seua informació. Aquests gestors no tenen flexibilitat en el disseny i les funcionalitats del portal, a més, compten amb moltes opcions que no serien útils per al aquest projecte. Això podria confondre a la persona que gestionarà el contingut, la qual no està familiaritzada amb les tecnologies de la informació i comunicació (TIC), per la qual cosa la corba d'aprenentatge per a aquesta persona potser elevada.

Per tant, la motivació d'aquest projecte és crear una aplicació web personalitzada, composta per un portal web on mostrar la informació i un gestor de contingut en el qual ella mateixa puga gestionar dita informació i portar la gestió del marxandatge d'una manera senzilla.

1.2. Objectius del Treball

En aquest treball existeix l'objectiu general establert per l'assignatura de consolidar el coneixement de l'estudiant relacionat amb el disseny web. No obstant això, per a desenvolupar el projecte, s'han de definir els objectius específics. Aquests objectius es poden dividir tant en objectius principals i objectius secundaris

1.2.1 Objectius principals

- Disseny i implementació d'un portal web funcional on mostrar la informació emmagatzemada
- Disseny i implementació d'un gestor de contingut
- Tant el portal web com el gestor de contingut han de ser adaptables a diferents tipus de dispositius.

1.2.2 Objectius secundaris

- Sistema de permisos basat en rols
- La interfície ha de ser senzilla d'utilitzar
- Ha de complir amb les regles d'accessibilitat

1.3. Impacte en sostenibilitat, ètic-social i de diversitat

En aquest apartat s'aborden i s'identifiquen els impactes del treball final en les dimensions de sostenibilitat, comportament ètic-social i diversitat. És important dur a terme aquesta anàlisi per a comprendre els efectes de la solució proposada.

En matèria de sostenibilitat, la solució d'aquest treball final intenta contribuir en la reducció de la problemàtica mediambiental existent. Com que els concerts s'anuncien per la plataforma web permet reduir la quantitat de cartells impresos promocionant l'esdeveniment i, per tant, estalviar i contribuir a la conservació de recursos naturals i la reducció de residus.

A més, s'ha eliminat l'opció de la venda en línia del marxandatge, substituint-la per la reserva prèvia i entrega en el mateix concert. D'aquesta manera es redueixen les emissions de CO₂ associades al transport de la mercaderia, s'eliminarà el trajecte de l'empresa que gestiona el transport fins al punt de recollida, del punt de recollida fins al punt d'entrega i la tornada a l'empresa.

Per un altre costat, el fet de reservar els productes, permet reduir el sobreestoc i optimitzar la producció, ja que es fabricaran i prepararan els productes segons el nombre de reserves, promocionant així la producció i consum responsable.

Si s'aprofundeix en els aspectes ètics i de responsabilitat social, cal comentar que l'aplicació intenta recollir el mínim nombre de dades personals de les persones registrades, només les estrictament necessàries per al bon funcionament d'aquesta. Sobre aquestes dades s'apliquen les Polítiques de privacitat, seguretat i protecció de dades segons la legislació vigent en matèria de protecció de dades i garantia dels drets digitals. Aquestes polítiques seran accessibles per a qualsevol persona que desitgi consultar-les.

Respecte al marxandatge es treballa amb empreses que en la seua producció treballen amb valors ètics, respectant els drets humans i lliure d'exploració laboral. Son empreses que, a més, impulsen el desenvolupament econòmic nacional creant llocs de treball en l'àmbit local i que afavoreixen la contractació de dones, reduint així la bretxa de desocupació.

Quan es parla de diversitat, és cert que comprén un ventall ampli de possibilitats, com aspectes de raça, religió, orientació sexual, funcional, etcètera. No obstant en aquesta anàlisi es centra en la diversitat funcional, ja que és l'únic aspecte que es veu reflectit en aquest treball, pel fet que no tracta cap dels altres temes. En aquest context, s'ha desenvolupat complint la norma WCAG2 per a reduir al màxim les barreres d'accessibilitat i garantir la igualtat d'accés per a totes les persones, independentment de les seues capacitats o condicions.

Si es parla de gènere, s'ha procurat crear un entorn equitatiu pel que s'ha pres la decisió de prescindir de descripcions de gènere, reconeixent l'existència de múltiples identitats i expressions. Tot el llenguatge de l'aplicació és inclusiu. Per exemple, en eliminar la distinció de gènere als productes, les samarretes deixen de ser d'home o dona i s'etiqueten com a samarretes entallades o rectes.

Per concloure aquest anàlisi es pot afirmar que aquest treball compleix distints Objectius de Desenvolupament Sostenible (ODS) en les tres dimensions proposades a la Competència de Compromís Ètic i Global (CCEG) . En la dimensió de sostenibilitat promou l'ODS 7 - Affordable and clean energy, l'ODS 12 - Responsible consumption and production i l'ODS 13 - Climate action. En la dimensió ètic-social es promou l'ODS 8 - Decent work and economic growth. I, per últim, en la dimensió de diversitat, gènere i drets humans l'ODS 5 - Gender equality i l'ODS 10 - Reduced inequalities

1.4. Enfocament i mètode seguit

El projecte plantejat en aquest treball final podria tindre dues estratègies per a dur-lo a terme, adaptar un producte existent o desenvolupar un producte nou.

En el cas d'adaptar un producte existent, hi ha plataformes com Wordpress, Blogger o Wix. A continuació, s'explica cada una d'elles:

- Wordpress: és una plataforma de gestió de contingut molt popular, ofereix una gran quantitat de plantilles i temes personalitzables i widgets. És possible crear des de blogs personals fins a comerços electrònics.
- Blogger: és una plataforma de blogs gratuïta, és utilitzada per a crear sobretot blogs personals. La interfície és senzilla d'usar, però la llibertat de personalització és molt limitada.
- Wix: és una plataforma de creació de llocs web amb plantilles predissenyades i un sistema d'arrossegar i soltar per al disseny del lloc web. És utilitzada per gent que vol tindre un lloc web però no té coneixements de programació.

Tots aquests productes permeten personalitzar plantilles i temes ja definits, a més d'un nombre de funcionalitats preestablertes que, potser, podrien satisfer

els requisits de l'artista. No obstant això, com que les plantilles són predefinides, no es compta amb total llibertat de disseny. Deixant de costat el cost en el cas de Wordpress i Wix per a eliminar anuncis en el lloc, es considera que les opcions i funcionalitats que s'utilitzarien són mínimes per al cost d'aprenentatge que suposa utilitzar una d'aquestes plataformes.

En el cas de desenvolupar un producte nou, cal crear un portal web i un gestor de contingut completament des de zero. D'aquesta manera, es garanteix que el lloc tindrà totes les funcionalitats i requisits que desitja l'artista. A més, comptarà amb total llibertat de disseny, proporcionant amb un aspecte únic i professional. Pot ser que aquesta opció siga més costosa, però com que es crearia un gestor de contingut des de zero, es redueixen el nombre de funcionalitats a les estrictament necessàries i així es redueix, per tant, la corba d'aprenentatge.

Finalment, s'ha optat per desenvolupar un producte totalment des de zero per garantir una personalització i control total de l'aplicació web. Permetent a l'artista gestionar el contingut des de la mateixa aplicació sense fer ús d'aplicacions externes, d'una manera ràpida i senzilla. Aquesta solució requereix més temps d'inversió, però a llarg termini oferirà un control i flexibilitat major.

Per a desenvolupar el producte des de zero, s'ha d'escollir una metodologia de desenvolupament. Dins de les distintes metodologies existents, s'han descartat les metodologies àgils, ja que aquestes metodologies no defineixen el projecte detalladament des de l'inici, sinó que estableixen unes bases i van replantejant el projecte segons van acabant els *sprints*, en un projecte amb un temps establert i limitat, aquesta metodologia és inviable. Com que aquest treball té uns requisits clars des de l'inici, marcats per la universitat, i una estructura seqüencial, s'ha optat per utilitzar la metodologia en cascada. En aquest mètode es realitzen una seqüència de passos, els quals no s'inicien fins a haver acabat l'anterior. Els passos principals d'aquesta metodologia són els següents.

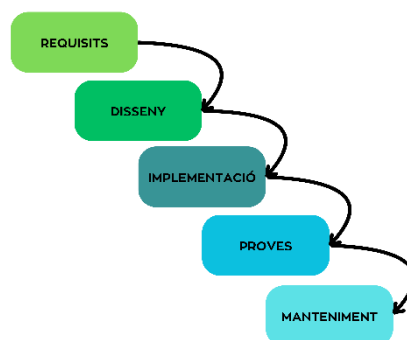


Figura 2: Metodologia de desenvolupament en cascada

- Requisites: En aquest punt es duen a terme un anàlisi detallat de les necessitats, requisits i objectius del projecte. Aquesta fase inclou la planificació de tot el procés de desenvolupament.
- Disseny: En aquest pas es dissenya l'estructura del sistema, on s'inclou la creació de diagrames, definició de tecnologies a utilitzar, casos d'ús i prototips d'alt i baix nivell.

- Implementació: Un cop acabat el disseny, es desenvolupen les funcionalitats del sistema mitjançant codi, cobrint tots els requisits i casos d'ús establerts durant la fase de disseny.
- Proves: En aquest punt es fan proves per assegurar que el codi desenvolupat compleix efectivament els requisits especificats i no conté errors.
- Manteniment: Una vegada finalitzat el treball i l'aplicació és posada en producció, s'inicia a la fase de manteniment, on es corregiran possibles errors que es puguin detectar en l'ús real de l'aplicació.

1.5. Planificació del Treball

El projecte, iniciat des de zero, es compon de quatre PAC tal com està establert per la UOC. Cada PAC ve marcada per un temps d'entrega, per a una millor planificació del treball, s'han dividit en tasques més específiques seguint la metodologia en cascada. La seua durada i ordenació estan reflectides en la següent figura.

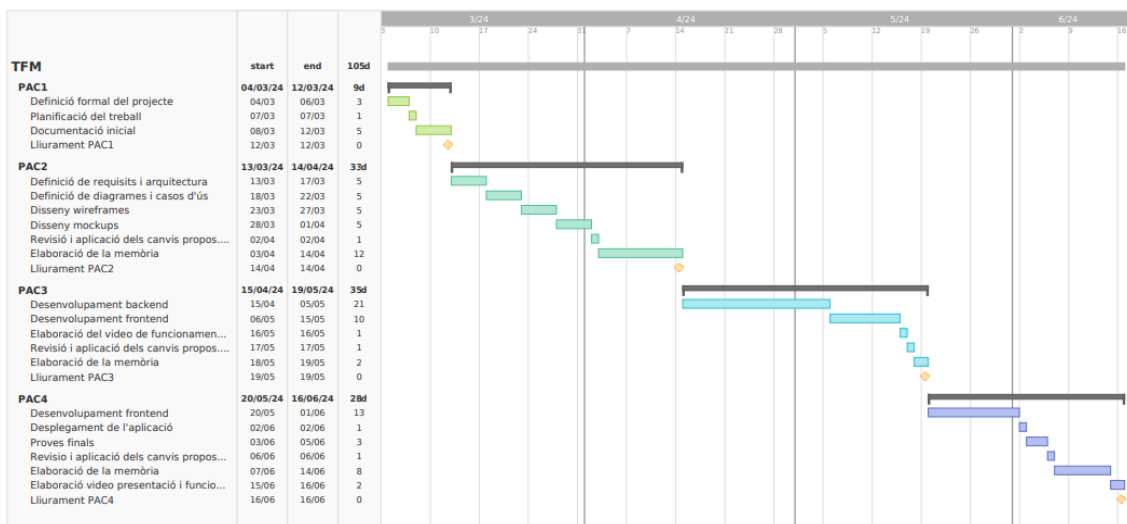


Figura 3: Diagrama de Gantt

1.6. Breu sumari de productes obtinguts

El producte resultant d'aquest projecte es tracta d'una aplicació web dividida en dues parts, la primera part es tracta d'un portal web públic on les persones podran registrar-se i iniciar sessió, consultar quins són els pròxims concerts i visualitzar productes de marxandatge disponibles. Una vegada iniciada la sessió podran reservar-los.

La segona part es tracta d'un gestor de contingut administrat per un sistema de rols. On totes les persones usuàries podran consultar les seues reserves. Aquelles amb perfil de gestor i administrador podran gestionar el contingut que es visualitza en el portal, és a dir, crear, editar i eliminar tant concerts com productes. Una altra funcionalitat de la qual disposen, és la de gestionar les comandes realitzades, filtrant-les per concerts i marcar si estan preparades i entregades.

Per acabar, aquelles persones amb perfil d'administradors podran llistar els i les usuàries registrades, modificar els seus perfils.

1.7. Breu descripció dels altres capítols de la memòria

Aquest projecte està organitzat en deu capítols de la manera següent.

En el segon capítol es definirà, dissenyarà el projecte. Aquest capítol es divideix en quatre blocs. En el primer bloc es definiran els requisits funcionals i no funcionals. Al segon es definirà l'arquitectura. Al tercer bloc, es dissenyarà l'aplicació definint els casos d'ús i els distints diagrames necessaris. I al quart bloc es dissenyaran els prototips d'alt i baix nivell.

El tercer capítol es detallarà el desenvolupament de la solució proposada. Aquest capítol dividirà en dos blocs, al primer es detallarà el desenvolupament del back-end i al segon el desenvolupament del front-end. Es justificaran els canvis duts a terme respecte a la previsió inicial del producte, en cas d'haver-n'hi.

En el quart capítol es definiran els tests realitzats, tant per al back-end com per al front-end.

En el capítol cinc, es realitzarà el desplegament de l'aplicació, es definiran els requisits i el desplegament tant per a un entorn de preproducció com per a producció.

En el sisé capítol es detallaran les instruccions d'ús per al portal públic i privat.

En el seté capítol s'informarà dels possibles errors que puga tindre l'aplicació a l'hora del lliurament i que no han pogut corregir-se.

En el capítol huit s'avaluarà si s'han complit els objectius plantejats a l'inici. A més, s'analitzarà si s'ha complit la planificació i en cas de fer ajustos, es justificarà. D'altra banda, es discutirà si s'han complit els objectius dels impactes de sostenibilitat, ètic-socials i de diversitat previstos.

En el capítol nou, es definiran els termes i acrònims més rellevants fets servir al llarg de la memòria, els quals ens facilitaran la comprensió del text.

Les referències utilitzades per a desenvolupar aquest projecte, es llisten en el capítol sis, fent servir la nomenclatura APA.

Per finalitzar en l'últim capítol es troben els elements complementaris per a entendre la memòria i el procés seguit en aquest treball.

2. Materials i mètodes

En aquest capítol es definirà, dissenyarà el projecte, en primer lloc, es durà a terme una anàlisi dels requisits, seguidament es definirà l'arquitectura de l'aplicació. En tercer lloc, es realitzarà el disseny d'aquesta i es crearan els prototips de baix i alt nivell.

2.1. Anàlisi de requisits

En aquest punt es detallen els requisits funcionals i no funcionals que s'han identificat per a l'aplicació web a desenvolupar. Aquests requisits, descrits des del punt de vista del sistema, permeten tindre una visió de les característiques que ha de tindre el projecte.

2.1.1. Requisits funcionals

RF01: Registre d'un/a usuari/ària

Les persones poden registrar-se a l'aplicació.

RF02: Autenticació

Permet a l'usuari o usuària obtenir un *token* propi introduint les credencials, el qual farà servir posteriorment per a fer les cridades al sistema.

RF03: Obtenir tots els concerts

Permet obtenir tots els concerts registrats a la persona usuària amb rol de gestor o administrador. I sols els pròxims sense necessitat d'iniciar sessió.

RF04: Obtenir concert per identificador

Permet obtenir un concert sense necessitat d'iniciar sessió.

RF05: Crear concert

Permet crear un concert a la persona usuària amb rol de gestor o administrador.

RF06: Actualitzar concert

Permet a la persona usuària amb rol de gestor o administrador actualitzar les característiques d'un concert.

RF07: Eliminar concert

L'usuari o usuària amb rol de gestor o administrador ha de poder eliminar un concert.

RF08: Obtenir tots els productes

Permet obtenir tots productes a la persona usuària amb rol de gestor o administrador i sols els disponibles sense necessitat d'iniciar sessió.

RF09: Obtenir productes per identificador

Permet obtenir un producte sense necessitat d'iniciar sessió.

- RF10: Crear producte
Permet crear un producte a la persona usuària amb rol de gestor o administrador.
- RF11: Actualitzar producte
Permet l'usuari o usuària amb rol de gestor o administrador actualitzar les característiques d'un producte emmagatzemat en la base de dades.
- RF12: Eliminar producte
La persona usuària amb rol de gestor o administrador ha de poder eliminar un producte.
- RF13: Obtenir totes les comandes
Ha de permetre a la persona usuària amb rol de gestor o administrador obtenir totes les comandes.
- RF14: Obtenir comandes per usuari
Ha de permetre a la persona registrada obtenir les comandes que ha realitzat.
- RF15: Crear comanda
Permet crear una comanda a la persona usuària.
- RF16: Actualitzar estat d'una comanda
Ha de permetre a la persona usuària amb rol de gestor o administrador actualitzar l'estat d'una comanda.
- RF17: Obtenir usuaris
La persona usuària amb rol administrador ha de poder obtenir els usuaris i usuàries registrades.
- RF18: Obtenir usuari
L'usuari o usuària registrada ha de poder obtenir les seues dades.
- RF19: Modificar rol d'un usuari
Ha de permetre l'usuari o usuària amb rol administrador actualitzar el rol d'un usuari.
- RF20: Actualitza dades d'un usuari
Permet a l'usuari o usuària actualitzar les dades del seu compte.
- RF21: Actualitza contrasenya d'un usuari
Permet a la persona usuària actualitzar la contrasenya del seu compte.
- RF22: Eliminar usuari
Permet a la persona usuària eliminar el seu compte.

2.1.2. Requisits no funcionals

- RNF01: Seguretat
Les dades restringides sols es podran consultar i modificar per les persones autoritzades. De la mateixa manera, les pàgines restringides

sols es podran consultar per persones autenticades en el sistema. Es comprovarà si la consulta o l'accés la realitza una persona autoritzada i en cas de ser necessari, el rol, abans de permetre l'operació sol·licitada. Els perfils seran bàsic, gestor i administrador.

RNF02: Usabilitat

L'aplicació ha de ser intuïtiva, senzilla d'utilitzar i ha de garantir l'eficiència d'ús. Permetent reduir el temps d'aprenentatge i finalitzar les tasques generant el menor nombre d'errors possibles.

RNF03: Rendiment

L'aplicació ha de ser capaç d'executar les tasques, processar les dades i respondre a la interfície d'usuari en un temps acceptable perquè l'experiència usuària siga fluida.

RNF04: Manejabilitat

L'aplicació ha d'estar estructurada de manera intuïtiva per tal que siga fàcilment modificable, per tal de garantir la manejabilitat del sistema i resoldre els possibles errors o l'ampliació d'aquest en un futur.

RNF05: Escalabilitat

El nombre de mètodes i funcionalitats s'ha de poder ampliar de forma senzilla i sense alterar l'estructura de l'aplicació i garantint l'escalabilitat.

RNF06: Protecció de dades

L'aplicació ha de tindre en compte el marc legal, ja que l'aplicació emmagatzema dades personals i sensibles de les persones registrades en la base de dades. Atesos a la Llei Orgànica de Protecció de Dades i garantia de drets digitals [2] s'ha de garantir la privacitat, confidencialitat i disponibilitat de les dades de la persona usuària.

2.2. Arquitectura de l'aplicació

En aquesta secció s'identifiquen els elements bàsics per a la implementació de l'aplicació. Als següents subcapítols s'identifiquen els components necessaris, l'arquitectura del sistema i es tenen en compte consideracions en l'àmbit de la seguretat i el rendiment de l'aplicació.

2.2.1. Identificació de components

Basant-se en els requisits descrits al capítol 2.1.1. Requisits funcionals, l'aplicació des pot dividir en mòduls, on cada mòdul agruparà els components relacionats.

Mòdul d'autenticació.

Agruparà els components de registre i autenticació. A més d'un component per a la gestió dels rols dels usuaris i usuàries registrades.

Mòdul de concerts.

Aquest mòdul contindrà els components que fan ús de les funcions necessàries per a gestionar els concerts. Es definirà un component concert on es mostraran les dades d'un únic concert, un concert-list on es llistaran diversos concerts. Un component gestió-concerts on es llistaran els concerts a gestionar i un component concert-form on s'agregaran o modificaran els concerts

Mòdul de productes

Agruparà els components que gestionen i consumeixen les dades relacionades amb els productes. Es definirà un component producte on es consumeixen les dades d'un únic concert, un producte-list on es llistaran els diversos productes. Un component gestió-productes on es llistaran els productes a gestionar i un component producte-form on s'agregaran o modificaran els concerts. Tindrà un component cistella, on llistaran els productes emmagatzemats al local-storage del navegador.

Mòdul de comandes

Aquest mòdul contindrà els components que usen les funcions referents a la gestió de les comandes. Un component confirma-comanda on es crearà una nova comanda. Un component comanda, on mostrarà les dades d'una comanda i es podrà modificar el seu estat. I un component comanda-list on llistarà les distintes comandes.

Mòdul d'usuaris

Aquest mòdul agruparà els components que interactuen amb les dades dels usuaris i usuàries registrades. Comptarà amb un component usuari que mostrarà, actualitzarà o eliminarà les dades de l'usuari o usuària registrada. I un mòdul usuari-list on es llistaran les dades de les persones usuàries registrades i on es podrà modificar el rol d'aquestes.

2.2.2. Arquitectura del sistema

L'aplicació, com es mostra a la figura 4, està formada per una arquitectura de tres capes, la capa de presentació o *front-end*, la capa de negoci o *back-end* i la capa de dades.

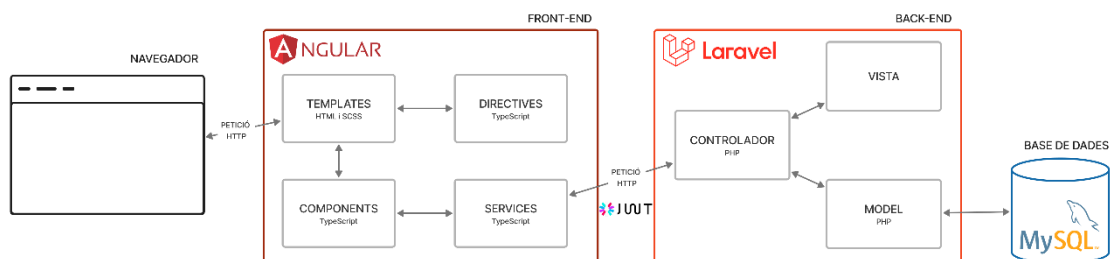


Figura 4: Diagrama d'arquitectura del sistema

La capa de presentació o *front-end*, està formada pel *framework* Angular. Aquest *framework* utilitza el model-vista-controlador i es programa amb TypeScript, HTML i SCSS. Serà l'encarregada de transmetre les peticions a la capa de negoci i mostrar les dades que aquesta retorne.

La capa de negoci o *back-end* està formada pel *framework* Laravel. Aquest *framework* també fa servir el model-vista-controlador, no obstant es programa amb PHP. Aquesta capa, proporcionarà una API REST que interaccionarà amb la base de dades. Aquesta capa rebrà peticions HTTP, obtindrà les dades de la base de dades i les proporcionarà a la capa de presentació.

La capa de dades empra com a sistema de gestió de base de dades relacional MySQL i és l'encarregada d'emmagatzemar les dades de l'aplicació.

2.2.3. Consideracions de Seguretat i Rendiment

Tant la seguretat com el rendiment, són dos aspectes fonamentals en el desenvolupament d'aplicacions web. Com s'ha vist al capítol 2.1.2 Requisits no funcionals, són dos punts a tindre en compte.

Per a garantir la privacitat dels recursos protegits cal utilitzar un sistema d'autenticació i autorització. Aquest requisit es pot complir fent un ús combinat de l'estàndard obert (RFC7519) JSON Web Token (JWT)[3] i les funcionalitats de seguretat d'Angular.

L'estàndard obert JSON Web Token permet transmetre informació de manera segura entre les distintes parts en format JSON. Quan un usuari o usuària s'autentica, es genera un *token* que consta de tres parts, la capçalera, la càrrega útil i la signatura. Seguint l'estructura "xxxxx.yyyyy.zzzzz".

La capçalera, codificada en Base64, indica quin tipus de *token* és i l'algorisme que s'usarà per a signar-lo, la càrrega útil conté la informació a transmetre i també es codifica en Base64. La signatura es genera fent servir una clau secreta, l'algorisme indicat, la capçalera i la càrrega útil, codificades ambdues.

Angular ofereix funcionalitats de seguretat com els interceptors HTTP[4] i les guardes de ruta[5]. Els interceptors HTTP permeten capturar les sol·licituds HTTP i modificar-les abans que arriben al servidor. Aquests interceptors permeten afegir capçaleres d'autorització entre altres funcions. Les guardes de ruta permeten controlar l'accés a les distintes rutes de l'aplicació i verificar si un o una usuària està autoritzada a accedir a una ruta en concret.

Per tant, combinant les funcionalitats, en autenticar-se l'usuari o usuària, generarà un JSON Web Token que s'emmagatzemarà al client. Aquest *token* s'inclourà en la capçalera de l'autorització de les sol·licituds que realitze usant interceptors HTTP d'Angular. Alhora, una vegada autenticat, fent servir les guardes de ruta d'angular es gestionaran les autoritzacions a les distintes pàgines segons el rol que tinga assignat. Quan es facen les sol·licituds, l'API rebrà el *token*, verificarà la signatura per a comprovar que no ha sigut alterat i extraurà la informació de la càrrega útil per a comprovar la identitat i els permisos

de l'usuari o usuària que ha dut a terme la petició. Si la informació és correcta, farà la petició a la base de dades.

Per a garantir l'òptim rendiment de l'aplicació, és important usar l'emmagatzemament local i la memòria cau per a reduir els temps de càrrega. Per un costat, l'emmagatzemament local permet emmagatzemar dades persistents d'ús recurrent per tal de no haver de consultar cada vegada al servidor. Per l'altre costat, emmagatzemar i consumir recursos de la memòria cau és possible fent ús d'una Progressive Web App (PWA).

Les Progressive Web App permeten, emmagatzemar la informació necessària per al bon funcionament de l'aplicació en la memòria cau. Quan es realitza una petició, mitjançant el Service Worker es comprova, en primer lloc, la memòria cau, reduint d'aquesta manera el temps de resposta. En cas de no trobar els recursos necessaris, consulta al servidor. A més, ofereix la capacitat de garantir el funcionament de l'aplicació inclús quan no es té connexió a la xarxa.

Per tant, combinant l'ús de l'emmagatzemament local i la memòria cau, junt amb els avantatges que ofereixen les PWA és possible millorar el rendiment i eficiència de l'aplicació.

2.3. Disseny de l'aplicació

En aquesta secció es dissenya la base de l'aplicació que guiarà el posterior desenvolupament. Als següents subcapítols es defineixen i detallen els casos d'ús, el sitemap de l'aplicació, els diagrames d'activitat i el model de dades.

2.3.1. Casos d'ús

En aquest punt es defineixen els casos d'ús que tindrà l'aplicació web. Com es veu a la figura 5, existeixen quatre tipus d'actors. L'usuari no registrat, l'usuari registrat, l'usuari gestor i l'usuari admin. L'actor usuari gestor estén les capacitats de l'actor usuari registrat. Alhora, l'actor usuari admin estén les capacitats de l'actor usuari gestor.

S'han definit set sistemes distints per a tindre agrupats els casos a partir dels mòduls identificats al capítol 2.2.1 Identificació de components. S'han definit els sistemes d'autenticació, concert, producte, comanda i usuari. No obstant això, s'ha definit el sistema portal web. Aquest engloba tots els casos d'ús exposats a l'exterior, és a dir, no formen part de la part privada de l'aplicació.

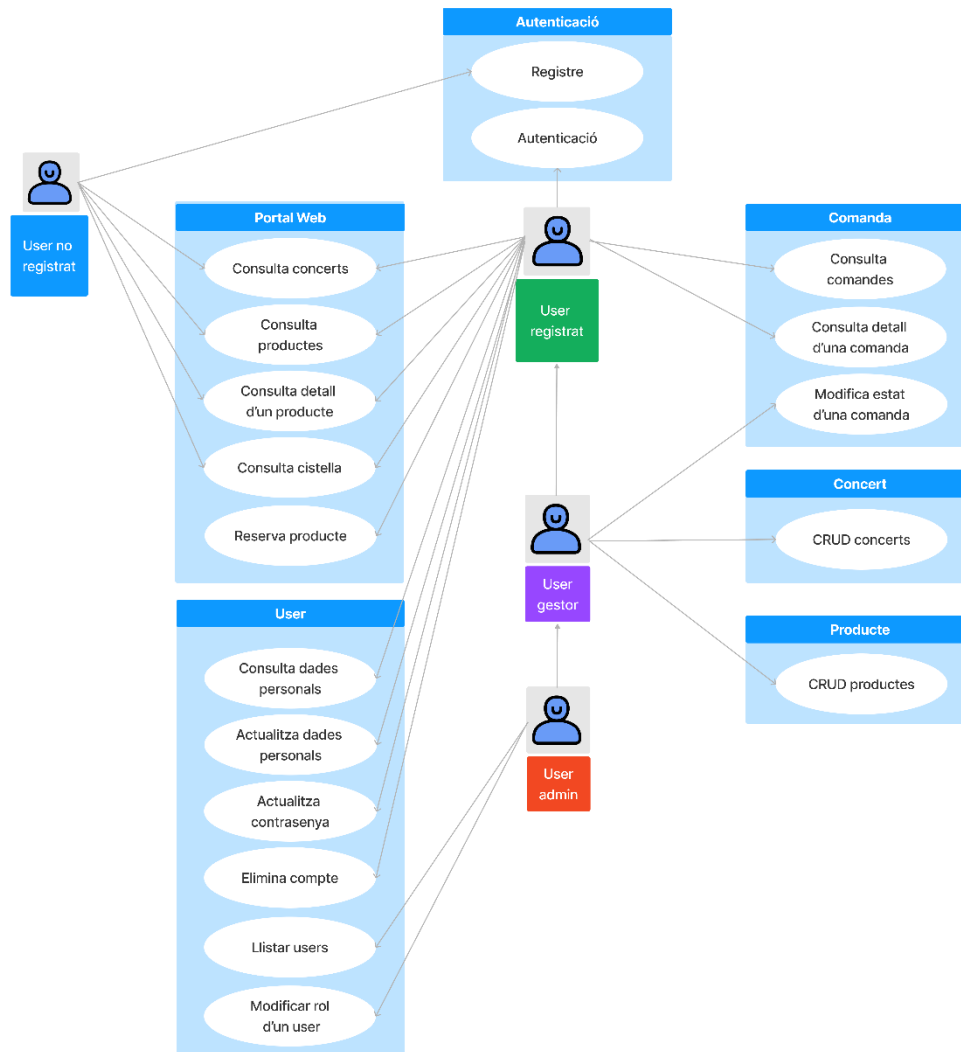


Figura 5: Diagrama de casos d'ús

A continuació, es detallen els casos d'ús del sistema autenticació.

Registre:

Per a accedir la primera vegada a la part privada de l'aplicació, la persona ha de registrar-se com a usuària.

Autenticació:

L'usuari o usuària registrada introdueix les seues credencials i el sistema comprova si són vàlides. En cas de ser-ho es genera un *token* únic.

El sistema portal web no requereix l'ús de tokens en cap cas d'ús, exceptuant en el cas Reserva productes. A continuació, es detallen cada un dels casos d'ús.

Consulta concerts:

Qualsevol usuari o usuària pot realitzar una petició per a obtenir els pròxims concerts.

Consulta productes

Qualsevol usuari o usuària pot realitzar una petició per a obtenir els productes disponibles.

Consulta detall d'un producte

Qualsevol usuari o usuària pot dur a terme una petició per a aconseguir les característiques d'un producte.

Consulta cistella

Qualsevol usuari o usuària pot fer una petició per a obtenir els productes emmagatzemats en la cistella.

Reserva productes:

Qualsevol usuari o usuària registrat pot reservar els productes de la cistella. El sistema comprova si el *token* proporcionat és vàlid.

El sistema comanda requereix l'ús de *tokens* i es comprovarà el rol de l'actor. A continuació, es detallen cada un dels casos d'ús.

Consulta comandes:

Qualsevol usuari o usuària registrat pot realitzar una petició per a obtenir les comandes. El sistema comprova si el *token* proporcionat és vàlid i si les comandes pertanyen a l'usuari o usuària. Si es tracta d'un usuari o usuària amb rol gestor, proporciona totes les comandes.

Consulta detall d'una comanda:

Qualsevol usuari o usuària registrat pot dur a terme una petició per a obtenir els detalls d'una comanda. El sistema comprova si el *token* és vàlid i si la comanda pertany a l'usuari o usuària, o si aquest té rol gestor.

Modifica estat d'una comanda:

Un usuari o usuària gestora pot fer una petició al sistema per a actualitzar l'estat d'una comanda. El sistema comprova si el *token* proporcionat és vàlid i el rol de l'usuari o usuària.

Tant el sistema concert, com el sistema producte es requereix l'ús de *tokens* i es comprovarà que el rol de l'actor siga gestor. A continuació, es detallen cada un dels casos d'ús.

CRUD Concert

Un usuari gestor pot realitzar una petició al sistema per a obtenir, crear, actualitzar o eliminar un concert. El sistema comprova si el *token* proporcionat és vàlid i el rol de l'usuari o usuària.

CRUD Producte

Un usuari gestor pot realitzar una petició al sistema per a obtenir, crear, actualitzar o eliminar un producte. El sistema comprova si el *token* proporcionat és vàlid i el rol de l'usuari o usuària.

Per acabar, al sistema usuari es requereix l'ús de *tokens* i es comprovarà que el rol de l'actor siga gestor o admin. A continuació, es detallen cada un dels casos d'ús.

Consulta dades personals

Qualsevol usuari o usuària registrat pot realitzar una petició al sistema per a obtenir les dades d'un usuari. El sistema comprova si el *token* proporcionat és vàlid i si les dades pertanyen a l'usuari o usuària registrada.

Actualitza dades personals

Qualsevol usuari o usuària registrat pot realitzar una petició al sistema per a actualitzar les dades d'un usuari. El sistema comprova si el *token* proporcionat és vàlid i si les dades a modificar pertanyen a l'usuari o usuària registrada.

Actualitza contrasenya

Qualsevol usuari o usuària registrat pot fer una petició al sistema per a actualitzar una contrasenya. El sistema comprova si el *token* proporcionat és vàlid i si la contrasenya a modificar pertany a l'usuari o usuària registrada.

Elimina compte

Qualsevol usuari o usuària registrat pot dur a terme una petició al sistema per a eliminar un compte. El sistema comprova si el *token* proporcionat és vàlid i si el compte pertany a l'usuari o usuària.

Llistar usuaris

Un usuari o usuària admin pot realitzar una petició al sistema per a obtenir un llistat amb els usuaris i usuàries registrades. El sistema comprova si el *token* proporcionat és vàlid i si el rol és admin.

Modifica rol d'un usuari

Un usuari o usuària admin pot realitzar una petició al sistema per a modificar el rol d'un usuari o usuària registrada. El sistema comprova si el *token* proporcionat és vàlid i si el rol és admin.

2.3.2. Sitemap

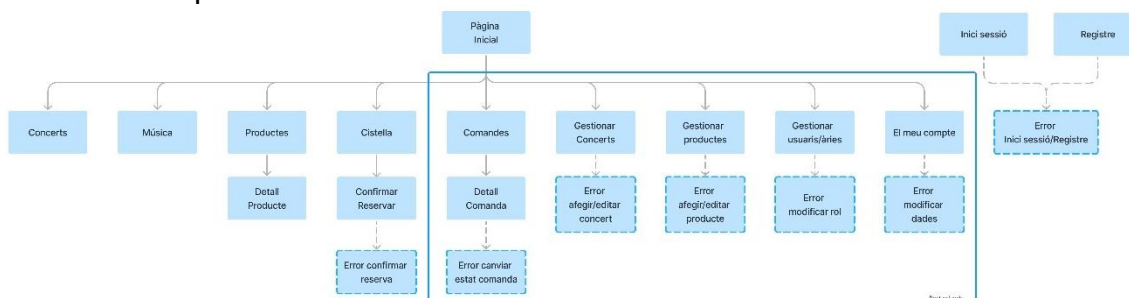


Figura 6: Sitemap de l'aplicació

Al Sitemap s'han creat les diferents opcions que tindran els menús de navegació principal i secundari, en el menú principal es troben les opcions públiques i al secundari les privades.

Des del menú públic, es pot accedir a les pàgines de concerts, música, productes i la cistella. Dins de la pàgina de productes es pot accedir a una pàgina addicional

amb els detalls del producte seleccionat i a la pàgina de la cistella és on s'emmagatzemaran les opcions de la comanda. Si es continua el procés de reserva, en cas d'haver iniciat la sessió prèviament, s'accedeix a la pàgina de confirmar reserva. S'ha tingut en compte una pantalla d'error a l'hora de confirmar la reserva, per si en algun moment el procés falla.

Des del menú privat es pot accedir a la pàgina de les comandes, la pàgina per a gestionar els concerts, la pàgina per a gestionar els productes, la pàgina per gestionar els usuaris/àries i la pàgina del perfil de l'usuari o usuària registrat.

La pàgina de les comandes permet llistar les distintes comandes que hi ha, siga per usuari, per concert o totes, segons el rol de l'usuari o usuària registrat. Des d'aquesta pàgina es pot accedir, en seleccionar una comanda, a una pàgina amb els seus detalls, si es té permís es podrà modificar l'estat d'aquesta. S'ha definit una pantalla d'error per si no es completa l'acció satisfactòriament.

A les pàgines gestionar concerts i gestionar productes es llisten els concerts i productes emmagatzemats a la base de dades. En aquestes dues pàgines és possible crear un element nou, modificar un existent o eliminar-lo sense necessitat d'accedir a una altra pàgina. S'ha definit una pàgina d'error per a afegir i editar un element nou i una altra per a quan s'elimina un element, aquestes pantalles apareixerien en cas de donar-se un error al procés.

La pàgina gestionar usuaris/àries permet llistar aquelles persones registrades en l'aplicació i modificar el seu rol sense haver d'abandonar aquesta pàgina. S'ha definit una pàgina d'error per si el procés no es completa correctament.

A la pàgina el meu compte es mostren les dades de la persona autenticada emmagatzemades a la base de dades. En aquesta pàgina és possible modificar les dades de contacte, la contrasenya o eliminar el compte sense necessitat d'accedir a una altra pàgina. S'ha definit una pàgina d'error per a editar les dades de contacte, una per al canvi de contrasenya i una altra per a quan s'elimina el compte, aquestes pantalles apareixerien en cas de donar-se un error al procés.

Com a pàgines addicionals hi ha l'inici de sessió i el registre en la plataforma, amb una pantalla d'error associada per si no es realitza satisfactòriament el procés.

2.3.6. Diagrames d'activitat

S'han dissenyat els diagrames d'activitat combinant el dels distintes casos d'ús agrupats pels sistemes definits i el sitemap.

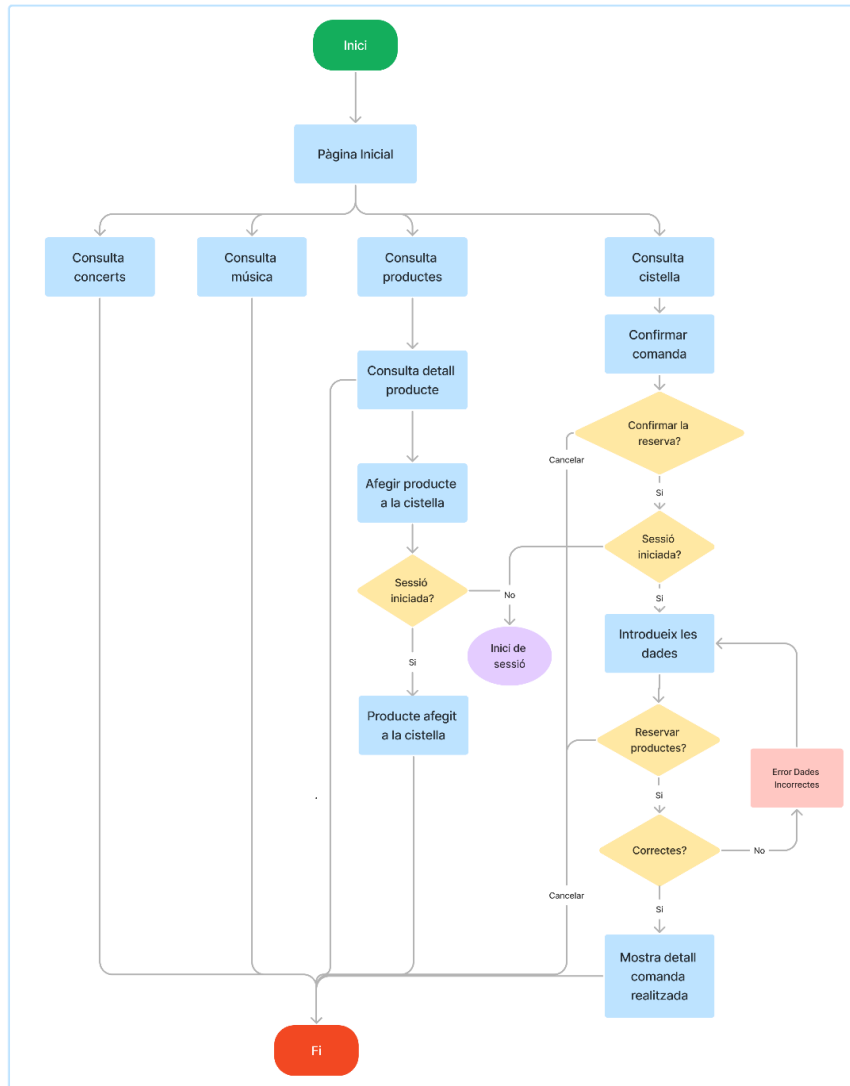


Figura 7: Diagrama d'activitat sistema portal web

En el sistema portal web, com es representa a la figura 7, l'usuari o usuària accedirà a la pàgina principal, i a partir d'ací podrà dur a terme quatre tasques diferents, consultar la pàgina dels concerts, consultar la pàgina de la música, consultar els productes o consultar la cistella.

En el cas de consultar els productes, podrà consultar el detall de cada un d'ells i afegir-los a la cistella posteriorment.

En el cas de consultar la cistella, podrà seguir el procés de reserva i confirmar la comanda. Si està la sessió iniciada, accedirà a la pàgina per a omplir les dades necessàries, si no està redirigirà la pàgina a l'inici de sessió. Una vegada introduïdes les dades i confirmada l'acció el sistema comprovarà si les dades són correctes. En el suposat, que les dades foren incorrectes es mostrarà un error, en cas contrari mostrarà el detall de la comanda duta a terme.

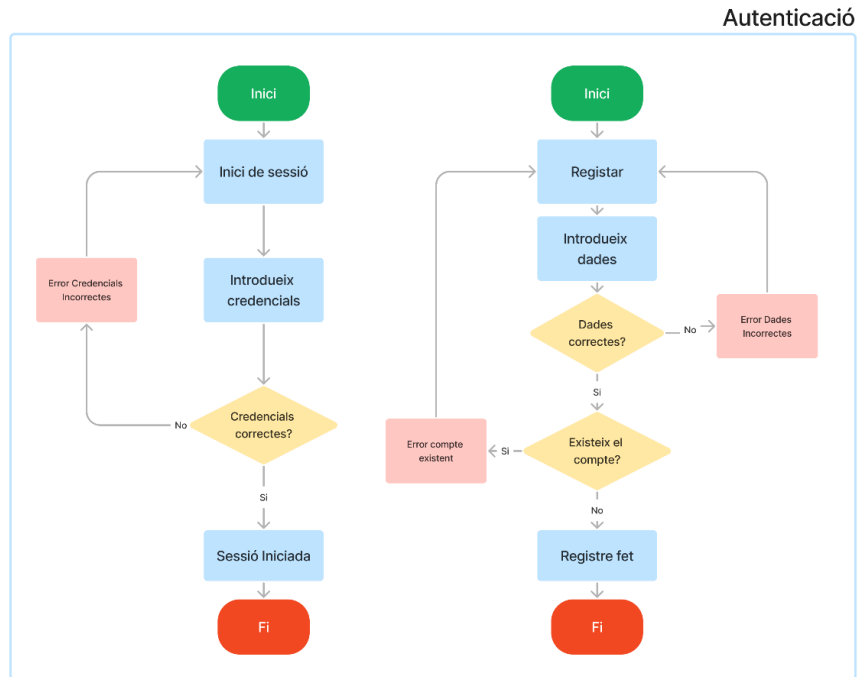


Figura 8: Diagrama d'activitat sistema d'autenticació

En el sistema autenticació, com es representa a la figura 8, l'usuari o usuària podrà dur a terme dues tasques independents, iniciar sessió o registrar-se. En cas d'accedir a la pàgina d'inici de sessió, si les credencials introduïdes són correctes, el sistema inicia la sessió. Si les credencials són incorrectes, es mostrarà un error.

En cas d'accedir a la pàgina de registre, l'usuari o usuària introduirà les dades i el sistema comprovarà si compleixen amb els requisits, en cas de complir-los, comprovarà si ja existeix un usuari amb les mateixes dades. En cas de no existir, registra l'usuari

Comanda

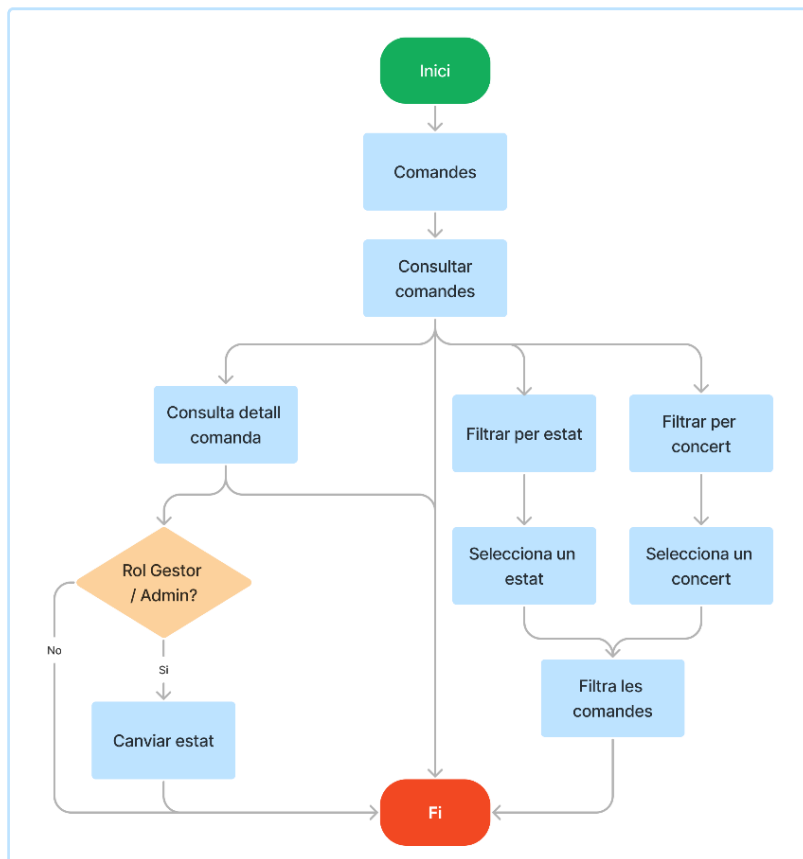


Figura 9: Diagrama d'activitat sistema comanda

En el sistema comanda, representat a la figura 9, l'usuari o usuària podrà accedir a la pàgina de comandes. Des d'aquesta pàgina podrà consultar totes les comandes realitzades, consultar el detall de cada una d'elles, filtrar-les per estat o per concert. Si el rol de l'usuari és gestor o administrador, podrà canviar l'estat de les comandes. Per a filtrar les comandes haurà de seleccionar un valor del filtre escollit i el sistema mostrarà els elements que corresponguen.

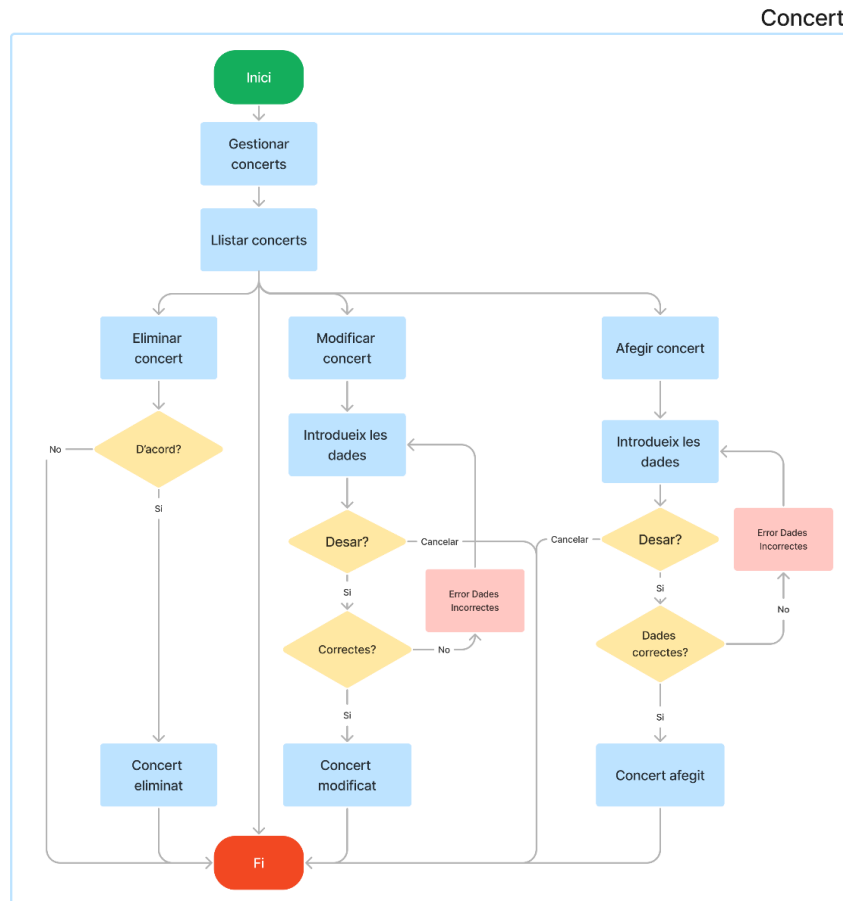


Figura 10: Diagrama d'activitat sistema concert

Al sistema concert, com es mostra a la figura 10, l'usuari o usuària, amb rol gestor o administrador, accedirà a la pàgina gestiona concerts. Aquesta pàgina mostrarà els concerts registrats a la base de dades i podrà eliminar, modificar o afegir concerts de nou.

En cas d'escollir eliminar un concert, el sistema demanarà confirmació abans de realitzar l'acció. Tant per a modificar, com per a afegir un concert nou, s'introduiran les dades, i es requerirà que es confirme l'acció. Si les dades no són correctes el sistema mostrarà un missatge d'error. En cas contrari es confirmarà i durà a terme l'acció.

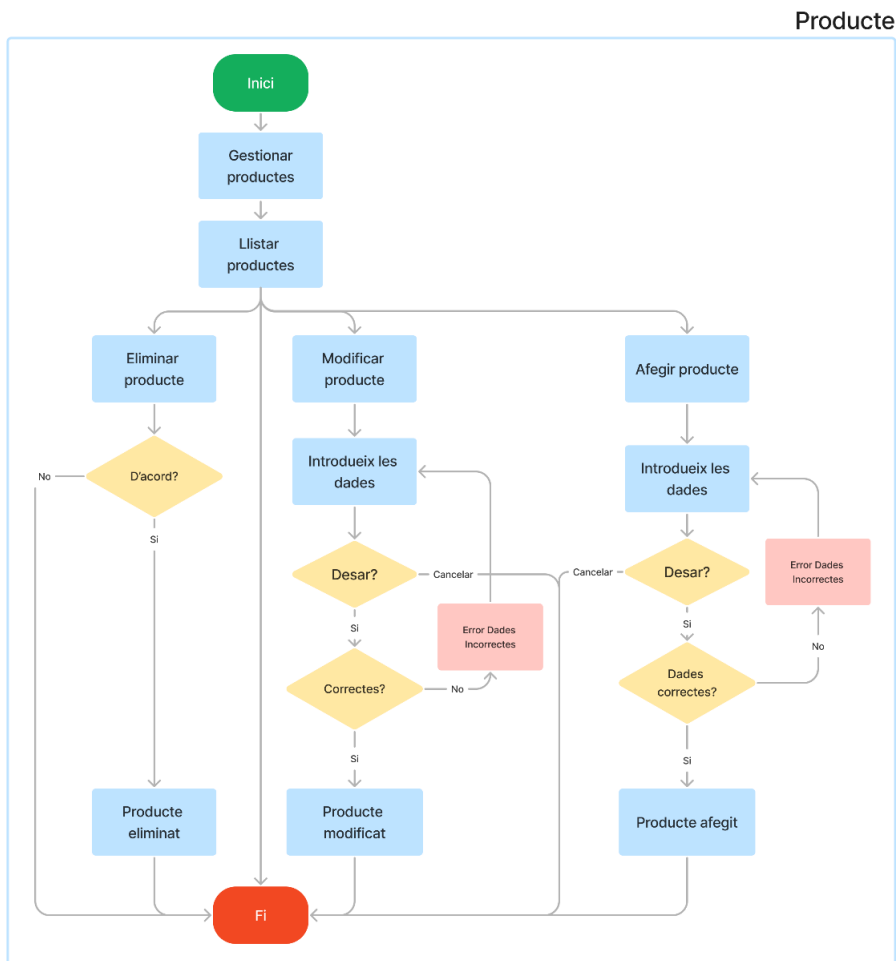


Figura 11: Diagrama d'activitat sistema producte

El sistema producte, com es mostra a la figura 11, té el mateix funcionament que el circuit anterior. L'usuari o usuària accedirà a la pàgina gestiona productes. Aquesta pàgina llistarà els productes emmagatzemats a la base de dades, i també es podran eliminar i modificar els productes existents o afegir-ne de nous.

En cas d'escollir eliminar un producte, el sistema demanarà confirmació abans de realitzar l'acció. En els casos de modificar i afegir un producte nou, s'introduiran les dades i caldrà confirmar l'acció perquè siga efectiva. Si les dades no són correctes el sistema mostrarà un missatge d'error. En cas contrari es confirmarà i portarà a cap l'acció.

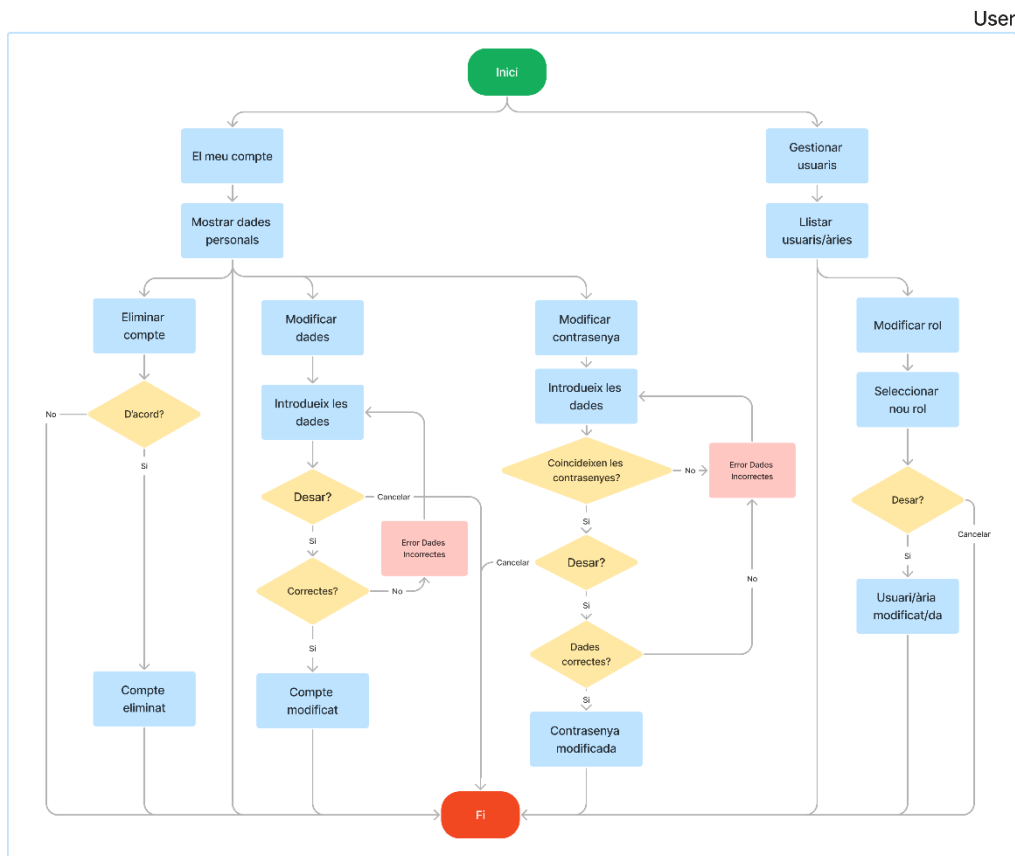


Figura 12: Diagrama d'activitat sistema usuari

En el sistema usuari qui estiga registrat i tinga els permisos corresponents podrà accedir a la pàgina el meu compte i a la pàgina gestionar usuaris.

Des de la pàgina el meu compte, es consulten les dades personals de la usuària o usuari autenticat. Es pot eliminar el compte, modificar les dades personals i la contrasenya. En cas d'escollir eliminar un producte, el sistema demanarà confirmació abans de realitzar l'acció. Per a modificar les dades personals o la contrasenya, s'introduiran les dades i caldrà confirmar l'acció perquè siga efectiva. Si les dades no són correctes, el sistema mostrarà un missatge d'error. En cas contrari es confirmarà i durà a terme l'acció. Per a canviar la contrasenya, el sistema farà la comprovació que la contrasenya introduïda i la de confirmació coincideixen. Si són iguals, el sistema deixarà confirmar l'acció, si no coincideixen el sistema mostrarà un missatge d'error i no deixarà confirmar l'acció.

Des de la pàgina gestiona usuaris, es llistaran tots els usuaris registrats al sistema. L'administrador o administradora podrà canviar el rol d'aquests seleccionant una opció i desant els canvis.

2.3.5. Model de dades

Com s'ha comentat al capítol 2.2.2. Diagrames d'arquitectura, les dades de l'aplicació s'emmagatzemaran en una base de dades relacional MySQL. A la figura 13 es mostra el diagrama d'ERD de l'aplicació.

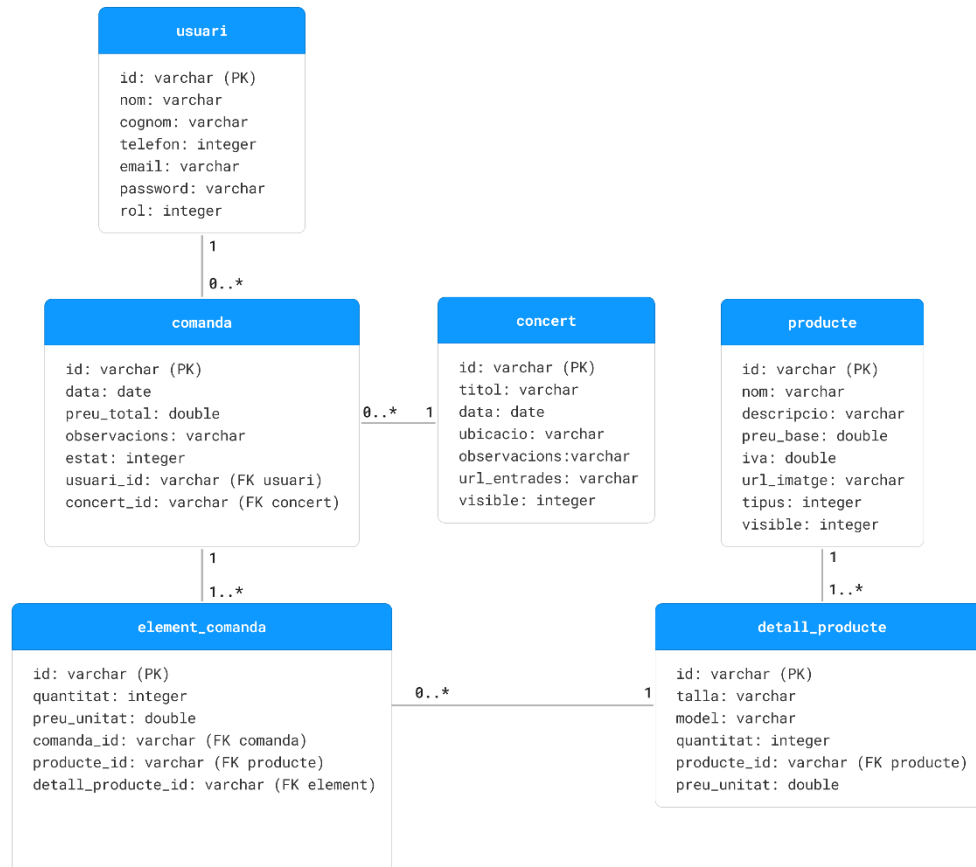


Figura 13: Model entitat-relació de l'aplicació

A continuació es detalla cada entitat i les relacions entre elles. Cada entitat té un identificador únic (PK) que permet relacionar-la amb altres entitats.

Usuari

Aquesta taula representa els usuaris i usuàries del sistema.

Comanda

Les comandes realitzades pels usuaris es registren en aquesta taula. Existeixen dues claus externes (FK), `usuari_id`, relacionada en el camp `id` de la taula `Usuari`, indica l'usuari que ha dut a terme la comanda i `concert_id`, relacionada en el camp `id` de la taula `Comanda`, indica a quin concert s'ha d'entregar.

Concert

Els concerts registrats al sistema s'emmagatzemen en aquesta taula.

Element_comanda

Aquesta taula relaciona la comanda amb el o els productes dels qual està composta. Existeixen tres claus externes (FK), `comanda_id`, relacionada en el camp `id` de la taula `Comanda`, indica a quina comanda pertany. LA clau `producte_id`, relacionada en el camp `id` de la taula `Producte`, indica quin producte conté i `detall_producte_id`, relacionada en el camp `id` de la taula `Detall_producte`, indica a el detall del producte que conté.

Producte

La informació general dels productes registrats al sistema s'emmagatzema en aquesta taula.

Detall_producte

Aquesta taula registra les variants i opcions de cada producte. Existeix una clau externa (FK), `producte_id`, relacionada en el camp `id` de la taula `Producte`, indica a quin producte pertany.

2.4. Prototipatge

En aquesta secció es dissenyen els prototips de baixa i alta fidelitat. Als dissenys es mostra l'aspecte que tindran les distintes pàgines definides al capítol 2.3.1 Sitemap. Com que l'aplicació web ha de ser adaptable, s'ha optat per dissenyar els prototips usant el mètode *mobile first*, on permet prioritzar la informació essencial i crear una base fàcilment adaptable per a dispositius més grans. Per tant, en els següents subcapítols es mostren les pàgines per a dispositius mòbils i en els casos on la diferència d'estructura és significativa, també per a dispositius de pantalla gran, de tipus escriptor. Aquests dissenys s'han realitzat amb la ferramenta Figma.

2.4.1. Wireframes

Els prototips de baixa fidelitat permeten veure el disseny de cada pantalla, la posició dels distintes elements, l'arquitectura de la informació i la navegació del lloc. En les següents imatges es mostren les pantalles amb un disseny esquemàtic i amb poc de detall de què serà posteriorment el producte finalitzat.

En primer lloc, es mostren les pantalles per l'autenticació, registre i recuperació de la contrasenya.

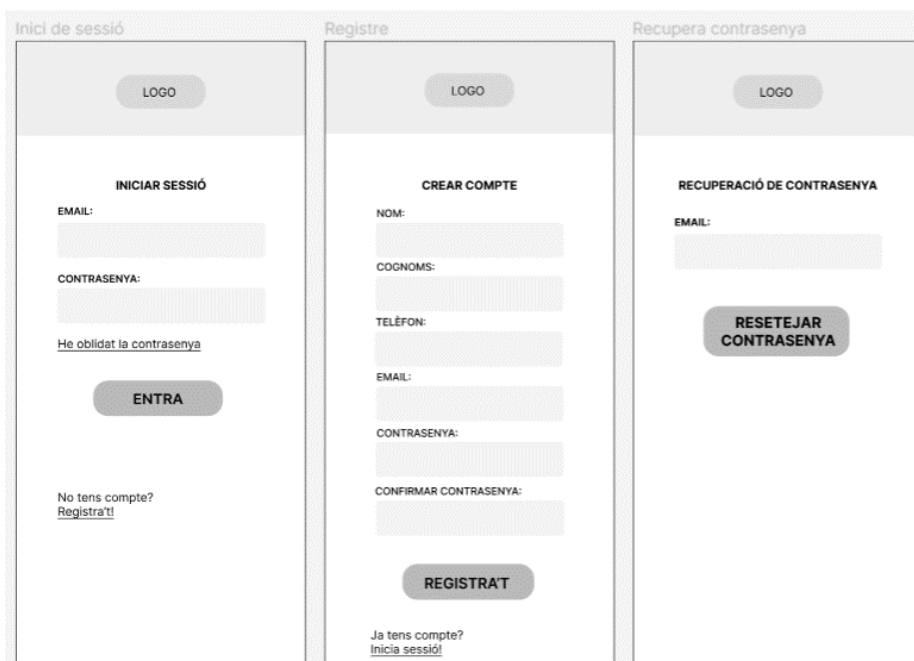


Figura 14: Wireframes d'autenticació, registre i recuperar contrasenya

En segon lloc, es troba la pantalla principal del portal web públic amb el menú desplegat, sense el menú desplegat i la pantalla de concerts.

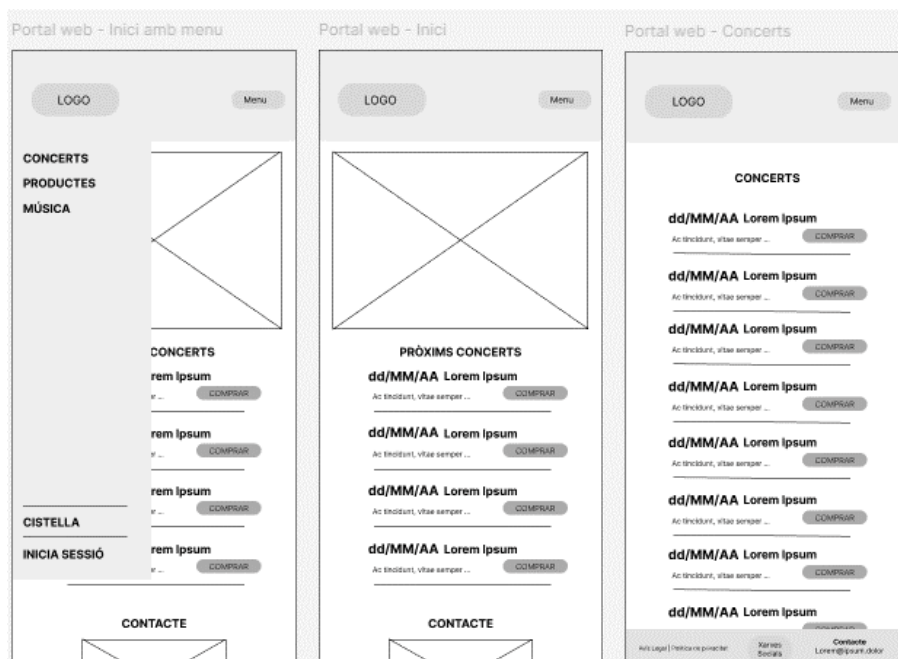


Figura 15: Wireframes pàgina d'inici amb menú, la pàgina d'inici i concerts

La pàgina de productes i música reestructuraran el seu contingut quan el dispositiu siga de pantalla gran, tal com es mostra en les següents figures.

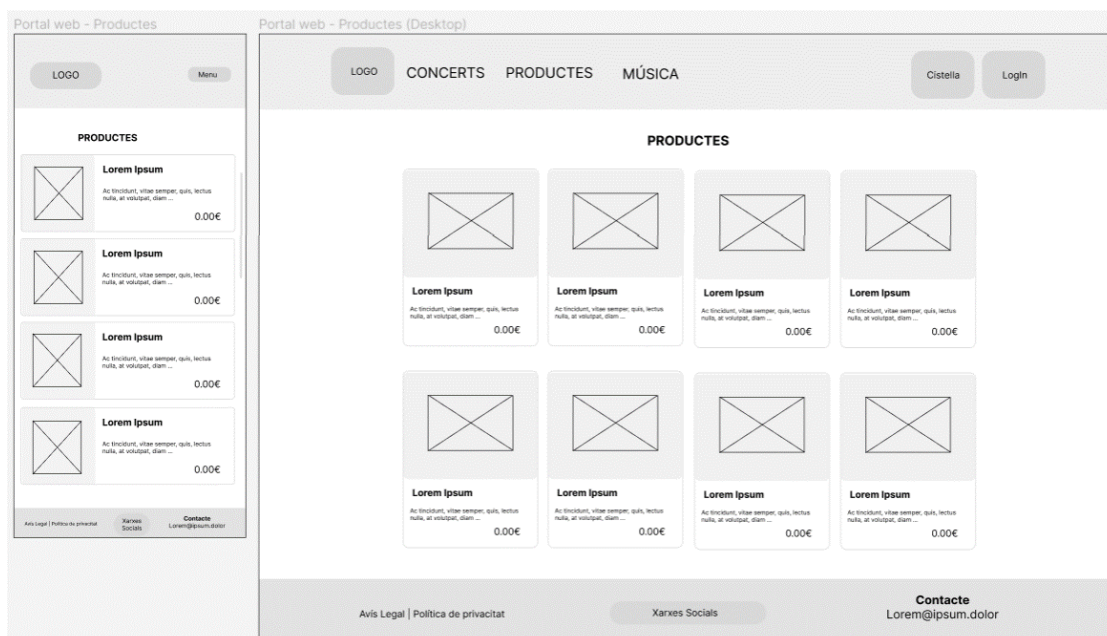


Figura 16: Wireframe productes versió mòbil i escriptori

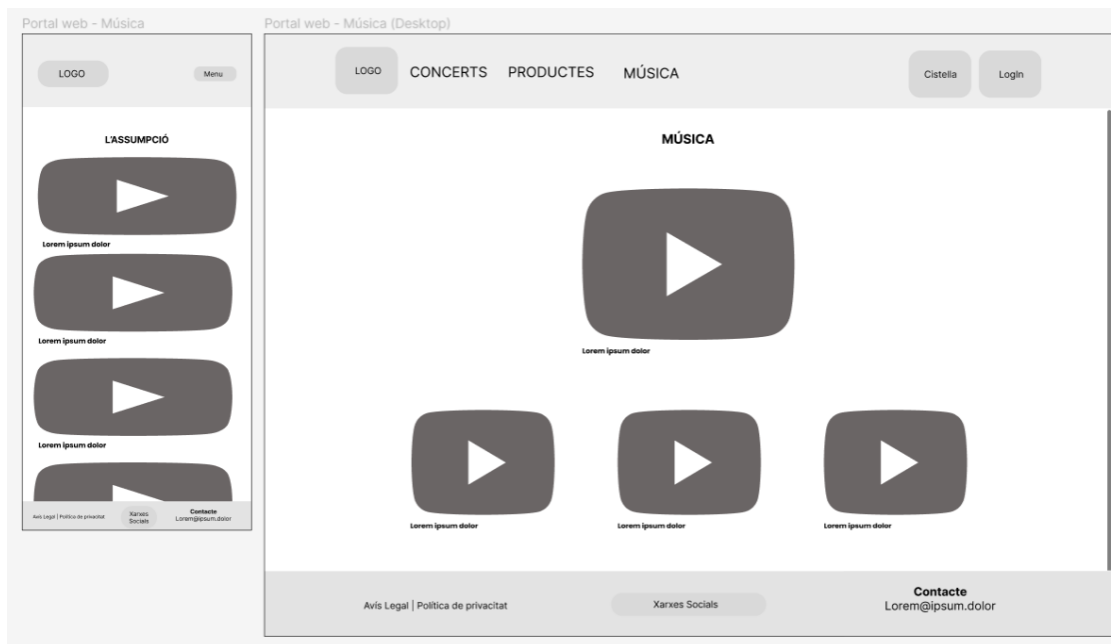


Figura 17: Wireframe música versió mòbil i escriptori

Seguidament, es mostren les pantalles del detall del producte, la cistella i la pàgina per a confirmar la reserva dels productes que hi ha a la cistella.

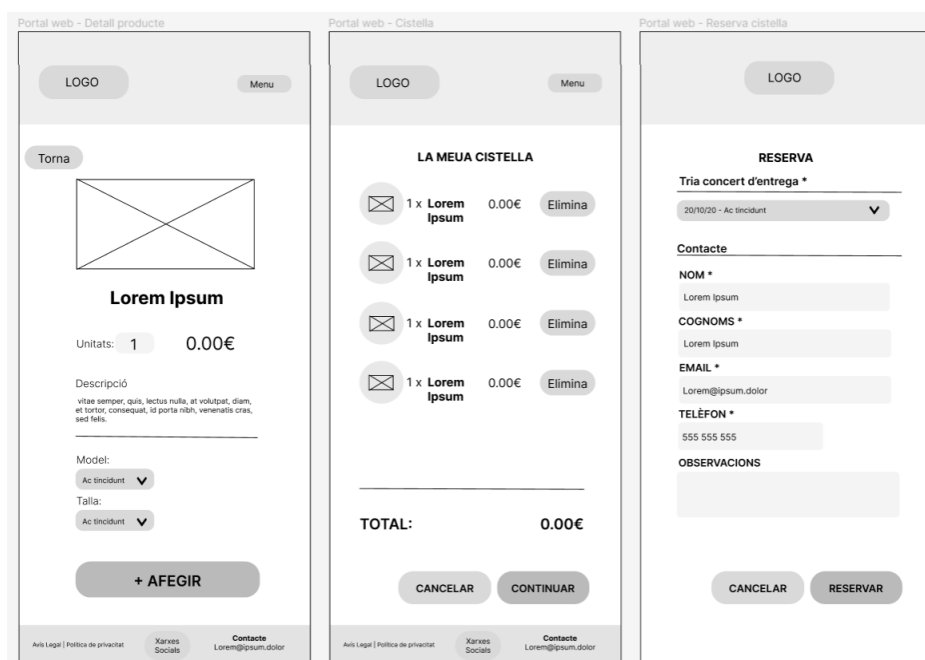


Figura 18: Wireframe pàgina detall del producte, cistella i reserva cistella

En tercer lloc, suposant que s'ha iniciat sessió, es mostren les pantalles de la part privada de l'aplicació web. Aquestes pantalles s'han creat suposant que ho visualitza un o una usuària amb el màxim de privilegis, és a dir, amb rol administrador.

Així doncs, el menú del portal privat es visualitzaria de la següent manera, en dispositius mòbils i en escriptori.

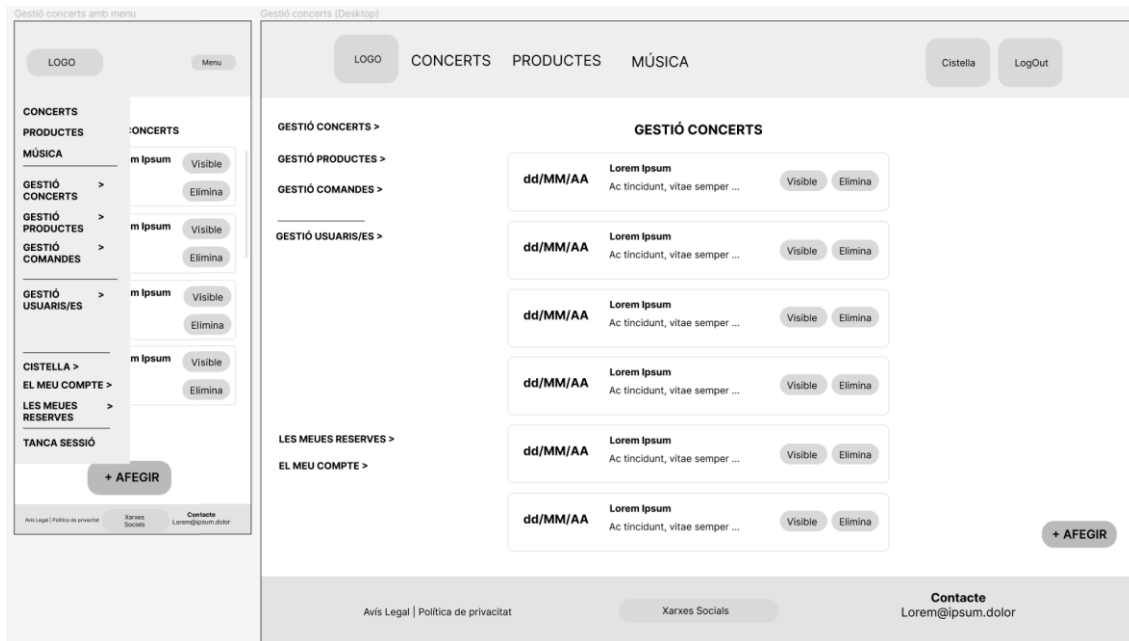


Figura 19: Wireframes menú versió mòbil i escriptori

La resta de les pantalles de gestió amb els seus respectius formularis es mostren a continuació.

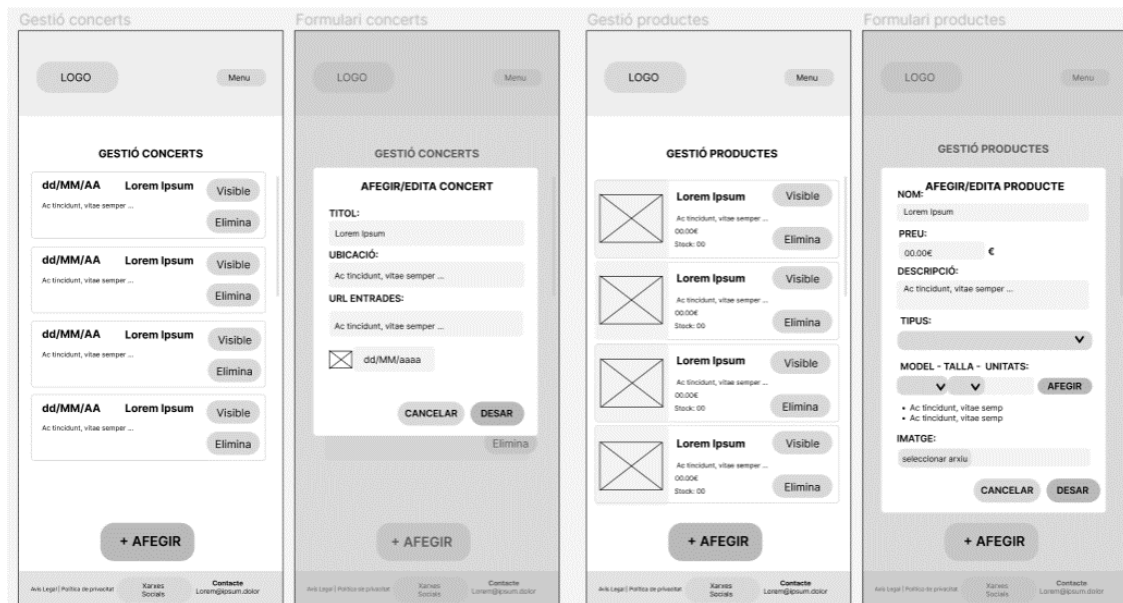


Figura 20: Wireframes gestió de concerts, gestió de productes i formularis

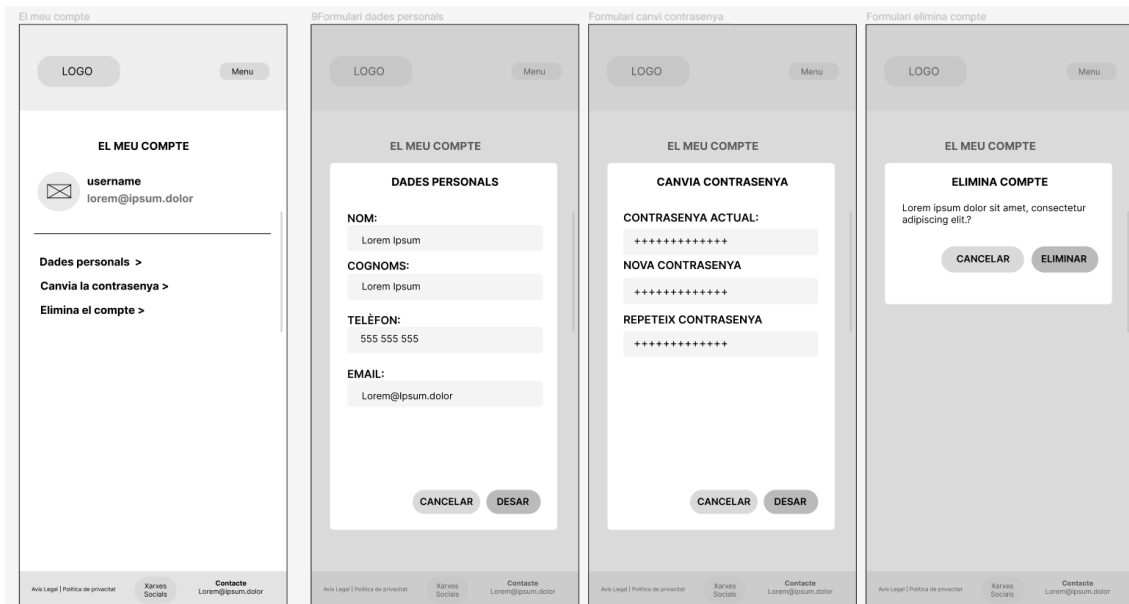


Figura 21: Wireframes el meu compte i formularis

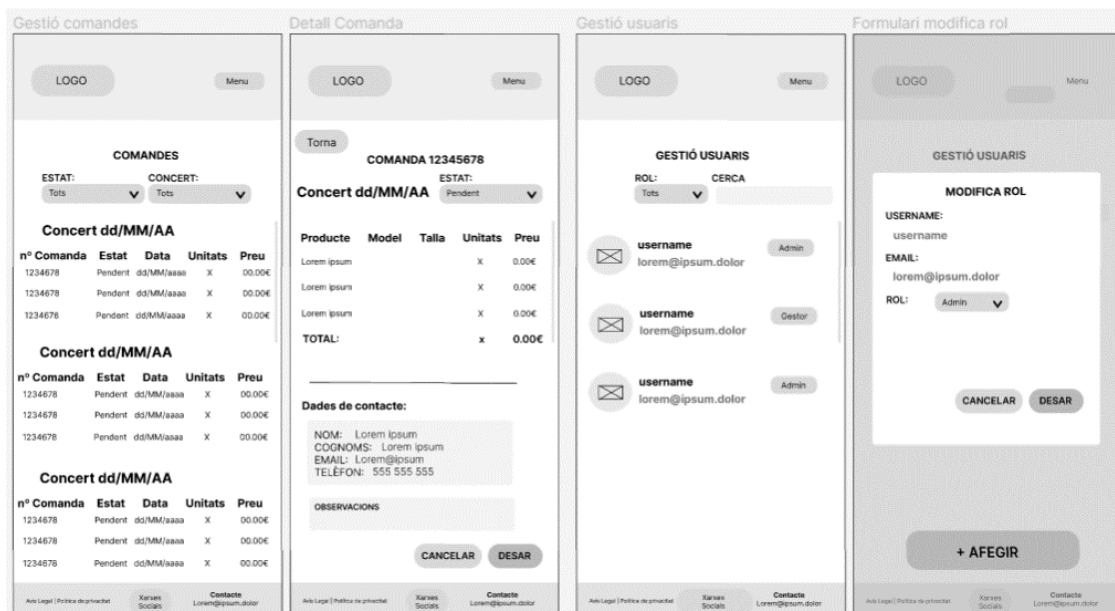


Figura 22: Wireframes comandes, detall comanda, gestió usuaris i formulari

2.4.2. Proposta d'interfície gràfica

Abans de dissenyar els prototips d'alt nivell o mockups, cal fer una proposta d'interfície gràfica per a mantindre una coherència i consistència en l'estil de l'aplicació.



Figura 23: Logotip de l'aplicació

Referent a la gamma cromàtica, s'ha optat per escollir una paleta de colors amb les mateixes tonalitats que el logotip de l'aplicació web. El qual s'observa en la figura anterior.

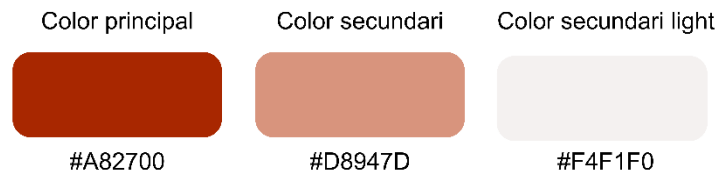


Figura 24: Paleta de colors de l'aplicació

A la figura 24, es defineix un únic color primari i dos colors secundaris. Aquests colors compleixen amb les directrius d'accessibilitat WACG2, en especial la directriu 1.4.6 Contrast (Millorat) a Nivell AAA [6] tal com es mostren en les figures 25, 26 i 27

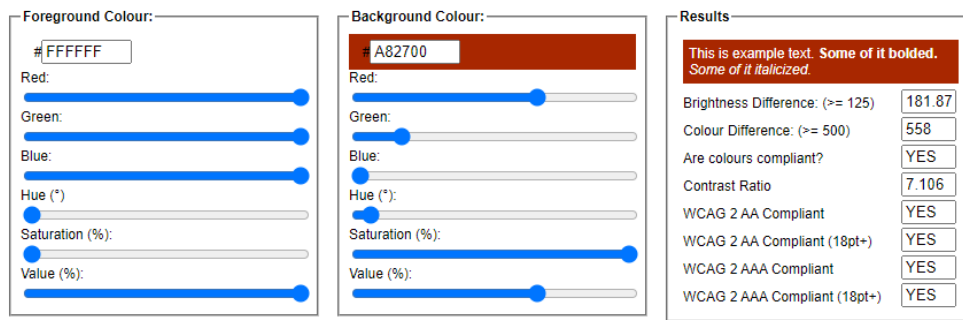


Figura 25: Comprovació de contrast del color principal

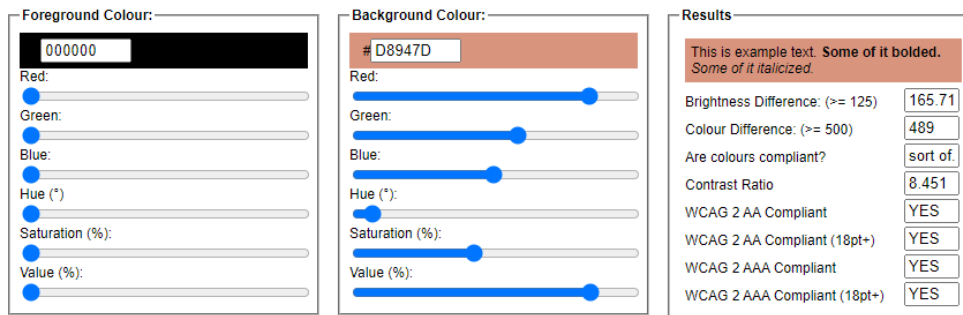


Figura 26: Comprovació de contrast del color secundari

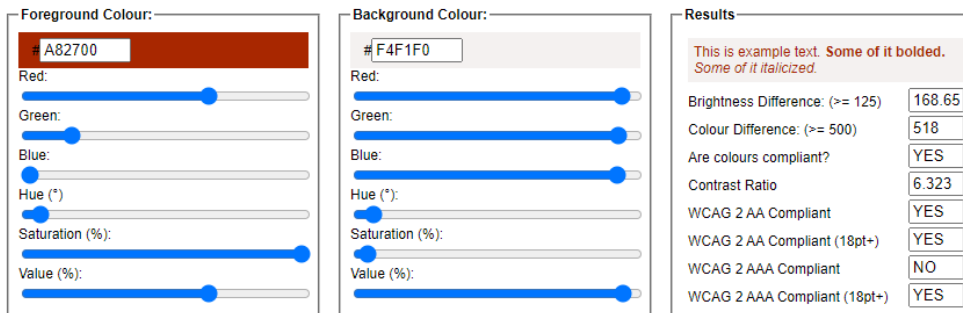


Figura 27: Comprovació de contrast del color secundari *light*

Respecte a la tipografia, tota l'aplicació web utilitza la font Poppins. El color d'aquesta dependrà del fons. A la capçalera i al peu de pàgina serà de color #FFFFFF, en canvi, al cos serà de color #000000, amb l'excepció de la data dels concerts, aquest text serà del color primari.

No tots els elements de l'aplicació web tindran el mateix estil. Els títols <h1> i els botons faran servir la font Poppins Bold, els subtítols <h2> i <h3> faran ús de Poppins Medium i el text pla i enllaços utilitzaran la font Poppins Regular.

Una vegada definida la gamma cromàtica i la tipografia, cal definir els estils dels elements interactius.

Per un costat, el fons dels botons *call-to-action*, la lletra i el marge dels botons primaris, els elements *checkbox* i les icones seran del color primari. I tant el text dels botons *call-to-action*, com el fons dels botons primaris seran de color #FFFFFF. Per l'altre costat, el fons dels elements input i *select* dels formularis seran del color secundari *light* amb el text negre. Finalment, els elements del menú desplegable i els botons, primaris i *call-to-action*, en estat *hover*, tindran el fons del color secundari. A la següent figura, es mostra l'aspecte que tindrà cada element i, a més, en el cas dels botons, l'estil que tindran en estat *hover*.

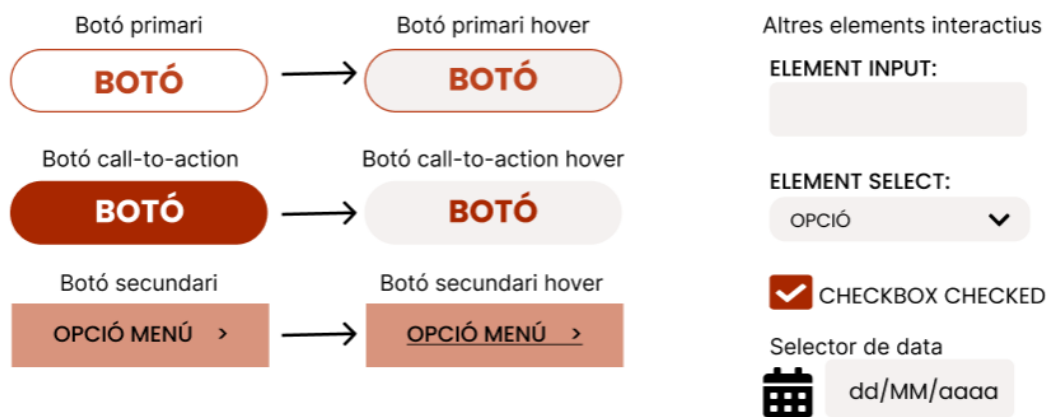


Figura 28: Estil d'elements interactius de l'aplicació

Les icones que s'utilitzaran mantindran sempre el mateix estil i mida. S'obtidran de la llibreria d'icones Font Awesome[7]. Es detallen a continuació.



Figura 29: Icones usades a l'aplicació

2.4.3. Mockups

Els prototips d'alta fidelitat o mockups permeten visualitzar quin aspecte tindrà el producte final d'una manera més realista. En aquests prototips inclouen elements gràfics, colors, tipografia de lletra, elements interactius i icones definits en la secció anterior.

S'han representat les mateixes pantalles que als prototips de baixa fidelitat. Per tant, s'han inclòs les pantalles amb versió escriptori representades anteriorment. En primer lloc, trobem les pantalles que representen les pàgines del portal web públic.

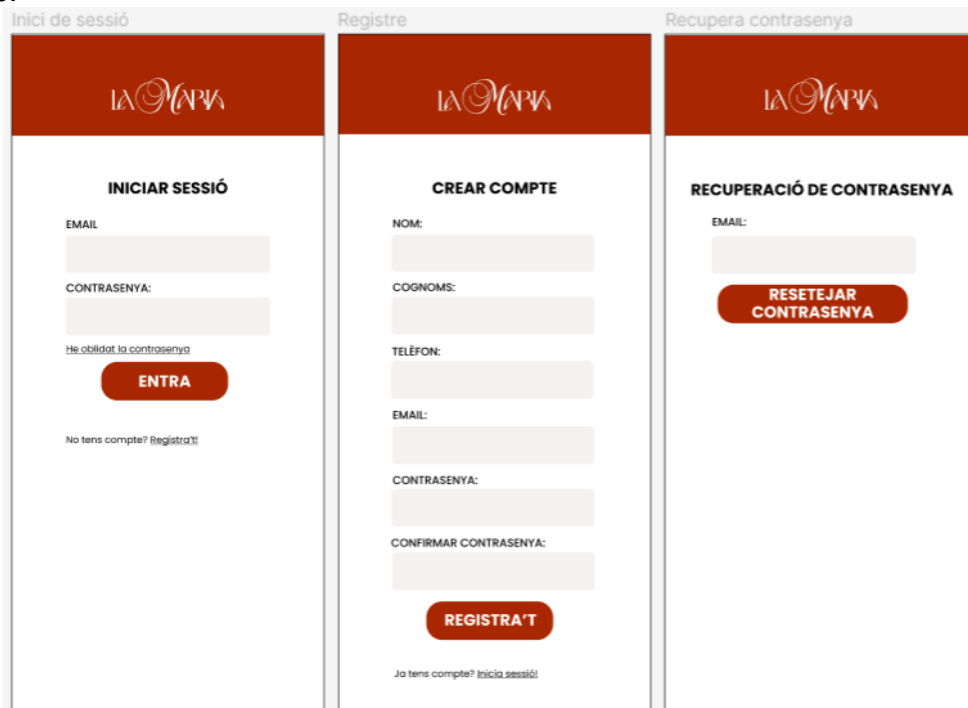


Figura 30: Mockups d'autenticació, registre i recuperar contrasenya

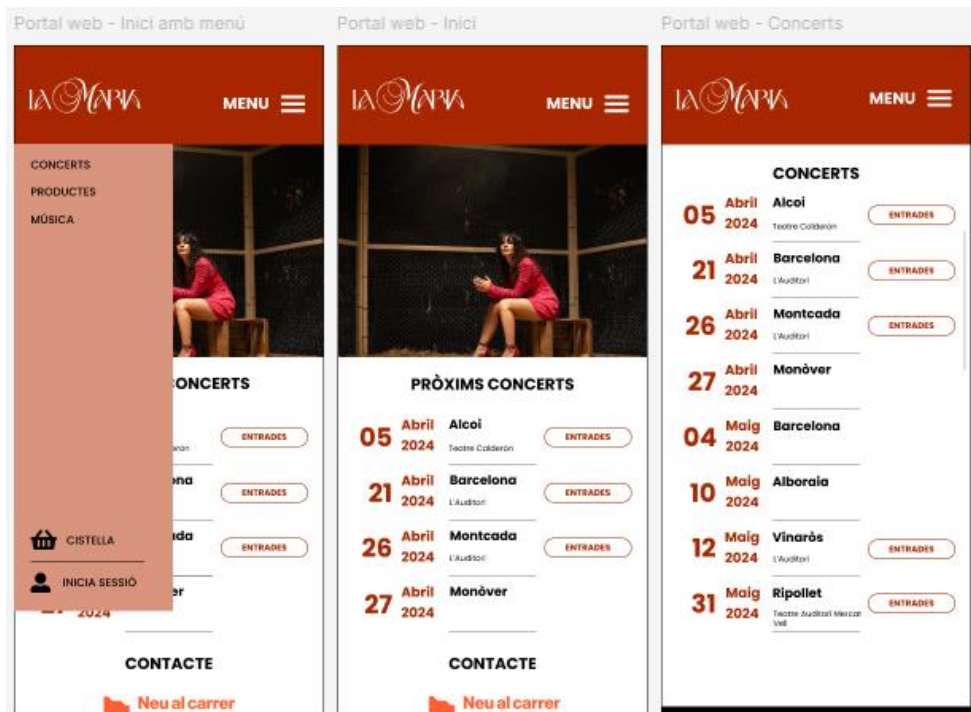


Figura 31: Mockups pàgina d'inici amb menú, pàgina d'inici i concerts [8][9]

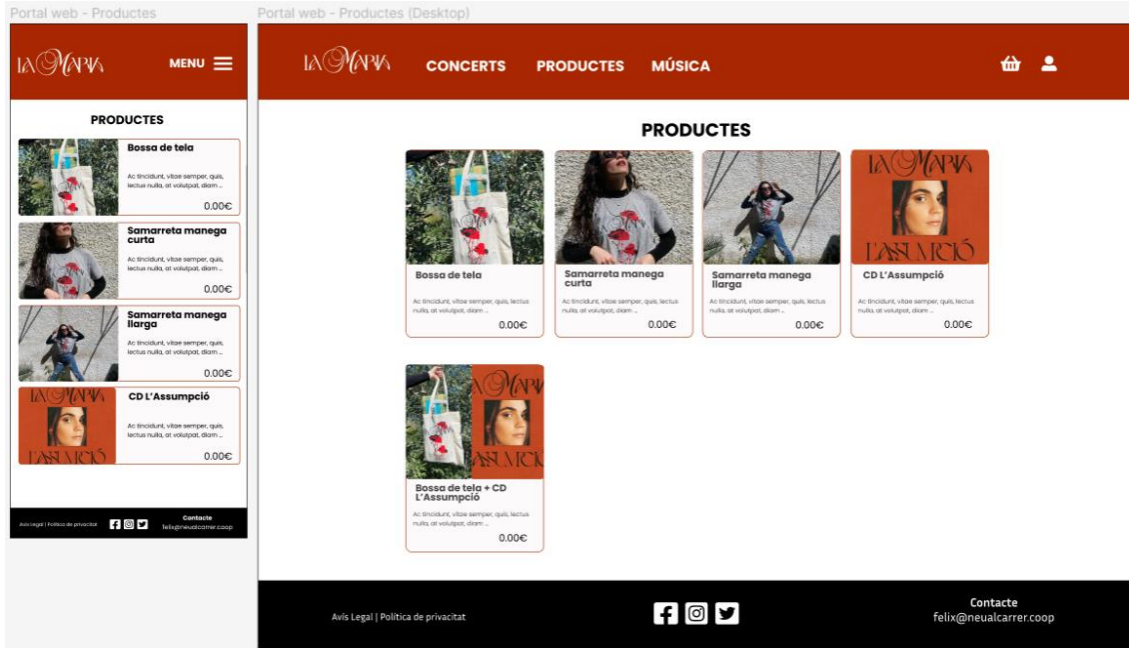


Figura 32: Mockups productes versió mòbil i escriptori [10][11]

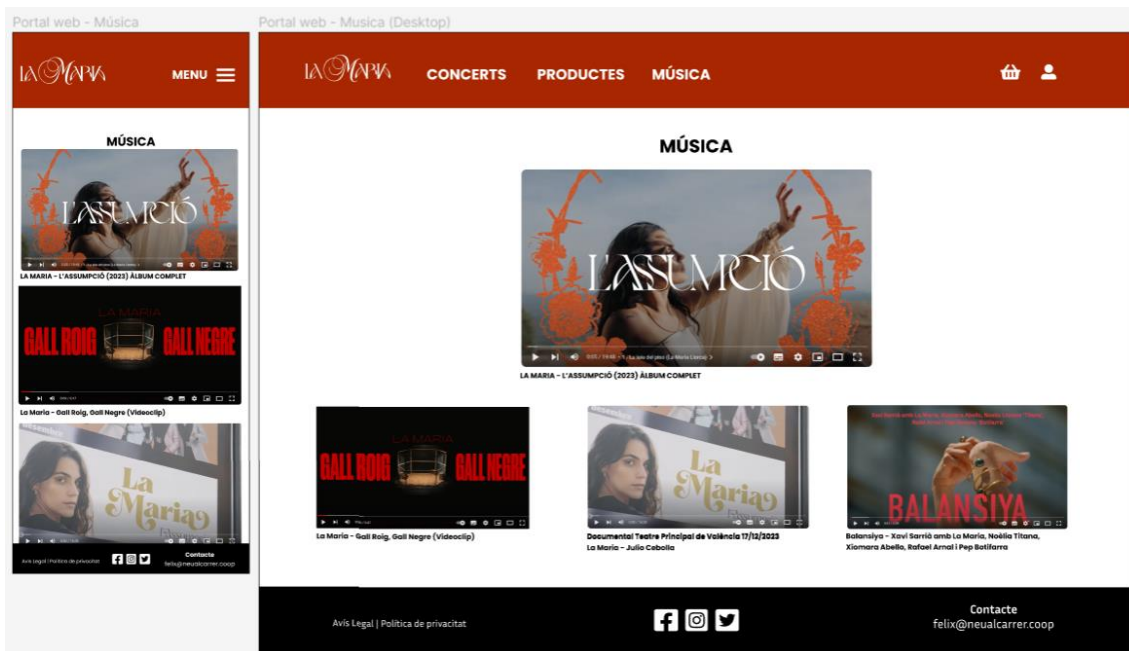


Figura 33: Mockups música versió mòbil i escriptori [12][13][14][15]

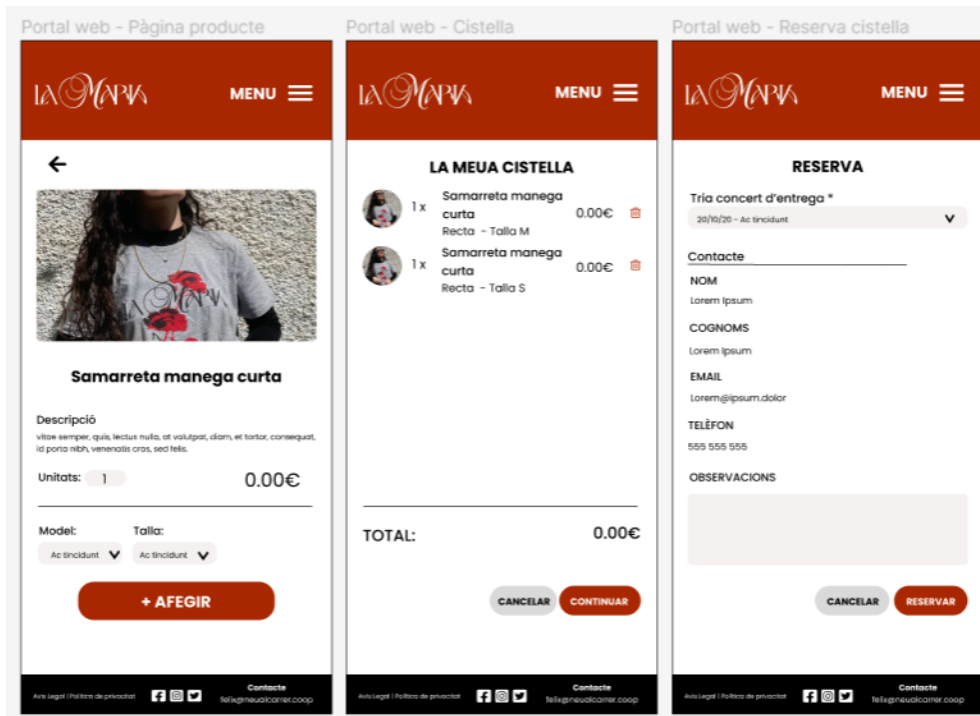


Figura 34: Mockups pàgina detall del producte, cistella i reserva cistella [10]

En segon lloc, es mostra el menú de navegació, una vegada iniciada la sessió amb un perfil amb alts privilegis, com el rol administrador.

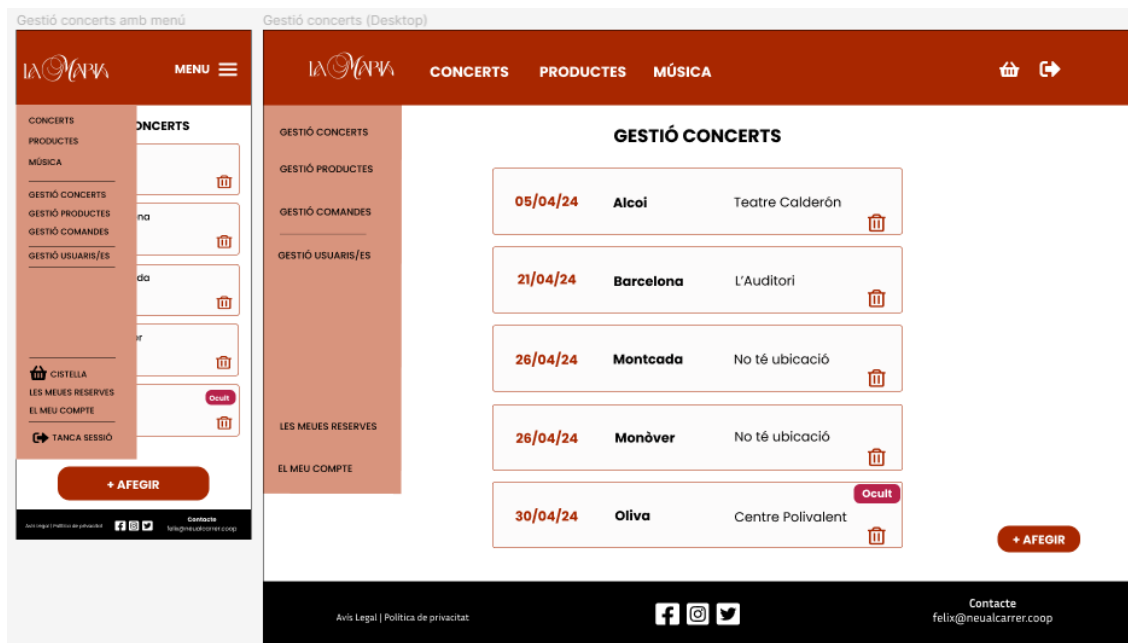


Figura 35: Mockups menú verisó mòbil i escriptori

En últim lloc, es representen les pantalles de la part privada de l'aplicació amb cada un dels seus formularis.

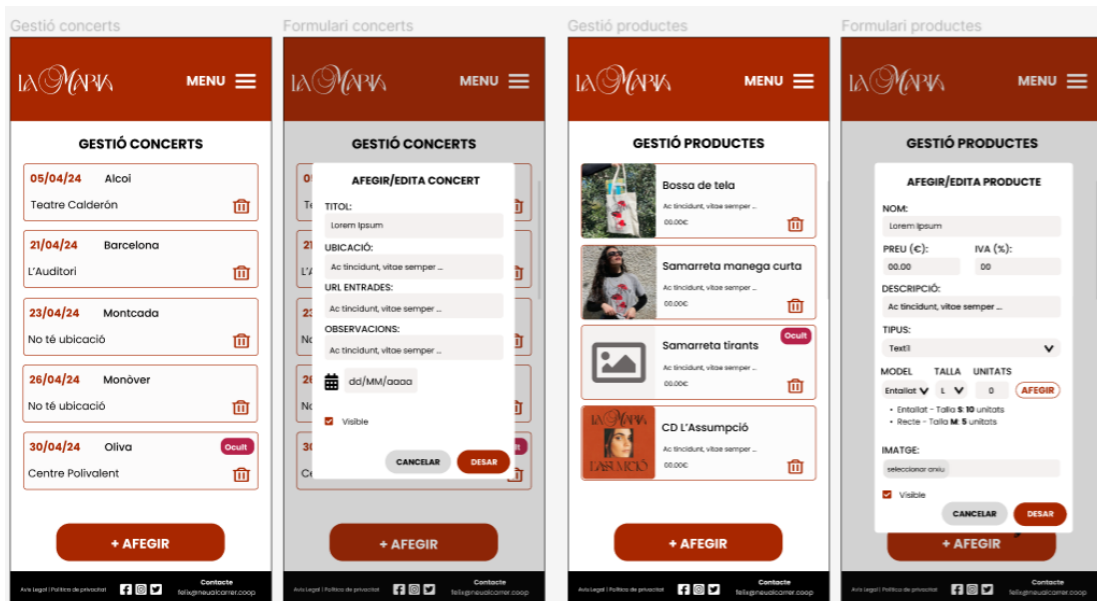


Figura 36: Mockups gestió de concerts, gestió de productes i formularis [10][11]

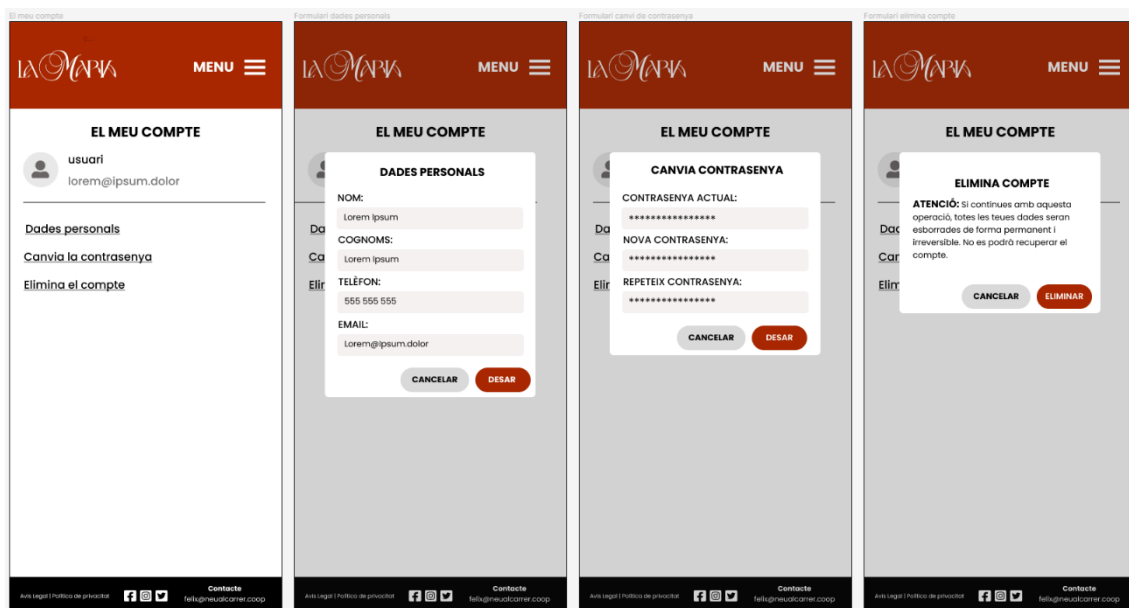


Figura 37: Mockups el meu compte i formularis

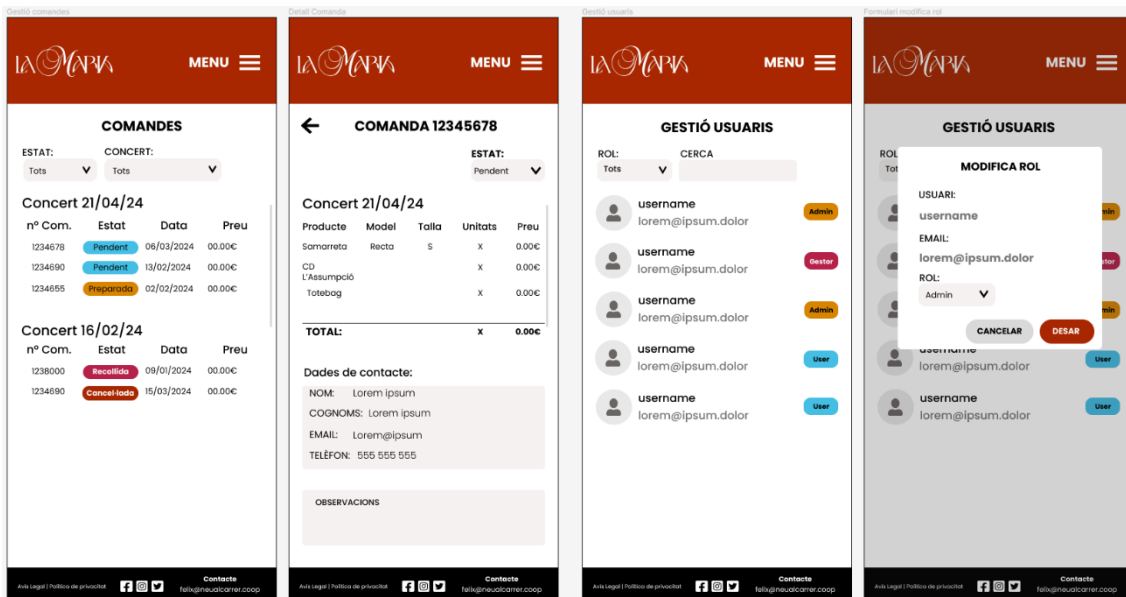


Figura 38: Mockups comandes, detall comanda, gestió usuaris i formulari

3. Desenvolupament

En aquest capítol es detallarà el desenvolupament de la solució proposada. En primer lloc, es veurà el desenvolupament de l'API REST i, en segon lloc, el desenvolupament de l'aplicació.

3.1. Back-End

L'API REST desenvolupada segueix l'estructura típica d'un projecte Laravel, la qual divideix el codi font i les configuracions en distints directoris.

El directori *app* conté la lògica de l'aplicació, ací s'han definit els controladors que manegen les sol·licituds HTTP, els *middleware*, els models i la classe auxiliar per a enviar correus electrònics.

Els controladors definits són els següents:

- AuthController: Es fa càrrec de manejar l'autenticació d'usuaris i interacciona amb la classe auxiliar per a enviar correus electrònics.
- UsuariController: S'encarrega de manejar les sol·licituds relacionades amb els usuaris.
- ConcertController: Maneja les sol·licituds relacionades amb els concerts.
- ProducteController: Maneja les sol·licituds relacionades amb els productes.
- DetallProducteController: S'encarrega de manejar les sol·licituds relacionades amb els detalls dels productes.
- ComandaController: Aquest controlador maneja les sol·licituds relacionades amb les ordres de compra.
- ElementComandaController: Es fa càrrec de manejar les sol·licituds relacionades amb els elements de les ordres de compra.
- ImageUploadController: Aquest controlador maneja la pujada d'imatges.

A l'annex 1 es mostra el funcionament de la funció encarregada de l'autenticació d'usuaris i el funcionament complet d'un dels controladors.

Els *middleware* definits restringeixen l'accés a les rutes que ho requereixen a tres nivells. A nivell d'usuari, de gestor i d'administrador.

- JWTMiddleware: Aquest *middleware* protegeix les rutes amb autenticació JWT, comprova que el JWT siga vàlid. D'aquesta manera sols els usuaris autenticats tenen accés a aquestes rutes.

```
try {
    $user = JWTAuth::parseToken()->authenticate();
    if (!$user) {
        return response()->json(['message' => 'No tens permís per a realitzar aquesta acció'], 401);
    }
} catch (JWTException $e) {
    return response()->json(['message' => $e->getMessage()], 500);
}
return $next($request);
```

Figura 39: Codi *middleware* JWT del *back-end*

- **GestorMiddleware:** Aquest *middleware* permet l'accés a les rutes a usuaris amb rol Gestor i Administrador. Comprova el rol en el JWT d'autenticació.

```

try {
    $rol = JWTAuth::getPayload()->get('rol');
    if (!$rol) {
        return response()->json(['message' => 'El rol no existeix'], 404);
    }
    if ($rol!="2" && $rol!="1") {
        return response()->json(['message' => 'No tens permís per a realitzar aquesta acció'], 401);
    }
} catch (JWTException $e) {
    return response()->json(['message' => $e->getMessage()], 500);
}
return $next($request);

```

Figura 40: Codi *middleware* Gestor del *back-end*

- **AdminMiddleware:** Aquest *middleware* permet l'accés a les rutes a usuaris amb rol Administrador. Comprova el rol en el JWT d'autenticació.

```

try {
    $rol = JWTAuth::getPayload()->get('rol');
    if (!$rol) {
        return response()->json(['message' => 'El rol no existeix'], 404);
    }
    if ($rol!="2") {
        return response()->json(['message' => 'No tens permís per a realitzar aquesta acció'], 401);
    }
} catch (JWTException $e) {
    return response()->json(['message' => $e->getMessage()], 500);
}
return $next($request);

```

Figura 41: Codi *middleware* Admin del *back-end*

Els models principalment defineixen amb quina taula de la base de dades han d'interaccionar els controladors.

El directori *database* conté els esquemes per a migrar les taules de la base de dades i, les llavors de dades (*seeders*) i fàbriques de models per a crear dades de prova per als usuaris, productes i concerts.

El directori *routes* conté el document *api.php* es defineixen les rutes com es mostra a la Figura 42, s'especifiquen els URL i el mètode de cada una d'elles. Les rutes amb els mètodes 'apiResource' indica que es crearan tots els mètodes bàsics d'una API REST. A més, s'assignen al controlador i s'indica els *middlewares* que intervindran.

```
Route::middleware(['jwt'])->group(function () {
Route::apiResource('usuari', UsuariController::class)->except('index','store');
Route::get('/usuari',[UsuariController::class, 'index']->middleware('admin');
Route::put('/usuari/{usuari}/password',[UsuariController::class, 'updatePassword'];
Route::put('/usuari/{usuari}/rol',[UsuariController::class, 'updateRol']->middleware('admin');
Route::apiResource('concert', ConcertController::class)->middleware('gestor');
Route::apiResource('producte', ProducteController::class)->middleware('gestor');
Route::apiResource('producte.detall', DetallProducteController::class)->middleware('gestor');
Route::apiResource('comanda', ComandaController::class)->except('update','destroy');
Route::put('/comanda/{comanda}',[ComandaController::class,'update']->middleware('gestor');
Route::apiResource('comanda.element', ElementComandaController::class)->except('update','destroy');
Route::post('/upload/{id}', [ImageUploadController::class, 'upload']->middleware('gestor');
});

Route::post('/login',[AuthController::class, 'login']->name('auth.login');
Route::post('/reset-password',[AuthController::class, 'resetPassword']->name('auth.resetPassword');
Route::post('/register',[UsuariController::class, 'store']->name('usuari.store');
Route::get('/concerts', [ConcertController::class, 'next']->name('concert.next');
Route::get('/disponibles', [ProducteController::class,'available']->name('producte.available');
Route::get('/disponibles/{producte}/detall', [DetallProducteController::class,'available']->name('producte.detall.available');
```

Figura 42: Codi definició rutes de l'API

El resultat de les rutes definides és mostra a la següent figura:

Mètode	URL	Controlador	Middleware
POST	api/login	AuthController	
POST	api/reset-password	AuthController	
POST	api/register	UsuariController	
GET	api/user	UsuariController	JWT, Admin
GET	api/user/{user}	UsuariController	JWT
PUT	api/user/{user}	UsuariController	JWT
DELETE	api/user/{user}	UsuariController	JWT
PUT	api/user/{user}/password	UsuariController	JWT
PUT	api/user/{user}/rol	UsuariController	JWT, Admin
GET	api/concerts	ConcertController	
GET	api/concert	ConcertController	JWT, Gestor
POST	api/concert	ConcertController	JWT, Gestor
GET	api/concert/{concert}	ConcertController	JWT, Gestor
PUT	api/concert/{concert}	ConcertController	JWT, Gestor
DELETE	api/concert/{concert}	ConcertController	JWT, Gestor
GET	api/disponibles	ProducteController	
GET	api/producte	ProducteController	JWT, Gestor
POST	api/producte	ProducteController	JWT, Gestor
GET	api/producte/{producte}	ProducteController	JWT, Gestor
PUT	api/producte/{producte}	ProducteController	JWT, Gestor
DELETE	api/producte/{producte}	ProducteController	JWT, Gestor
GET	api/disponibles/{producte}/detall	DetallProducteController	
GET	api/producte/{producte}/detall	DetallProducteController	JWT, Gestor
POST	api/producte/{producte}/detall	DetallProducteController	JWT, Gestor
GET	api/producte/{producte}/detall/{detall}	DetallProducteController	JWT, Gestor
PUT	api/producte/{producte}/detall/{detall}	DetallProducteController	JWT, Gestor
DELETE	api/producte/{producte}/detall/{detall}	DetallProducteController	JWT, Gestor
POST	api/upload/{id}	ImageUploadController	JWT, Gestor
GET	api/comanda	ComandaController	JWT
POST	api/comanda	ComandaController	JWT
GET	api/comanda/{comanda}	ComandaController	JWT
PUT	api/comanda/{comanda}	ComandaController	JWT, Gestor
GET	api/comanda/{comanda}/element	ElementComandaController	JWT
POST	api/comanda/{comanda}/element	ElementComandaController	JWT
GET	api/comanda/{comanda}/element/{element}	ElementComandaController	JWT

Figura 43: Rutes de l'API amb controlador i *middlewares* assignats

Cal comentar que un canvi important respecte a la planificació inicial que s'ha realitzat en el *back-end* és l'acció d'eliminar un usuari. Com que la taula comandes té una clau externa de la taula usuaris, si aquest té comandes

realitzades, en lloc d'eliminar l'usuari en la base de dades, eliminen les dades administratives i es canvien per l'identificador de l'usuari.

```

/**
 * Elimina les dades de l'usuari
 */
public function destroy(string $id)
{
    $usuari = Usuari::findOrFail($id);
    $usuari->update(
        [
            'nom'=>$id,
            'cognoms'=>$id,
            'telefon'=>0,
            'email'=>$id,
            'password'=>$id.$id,
            'rol' => -1
        ]);
    return response()->json(200);
}

```

Figura 44: Funció elimina usuari del back-end

D'aquesta manera es pot realitzar un control de les comandes per part de les persones gestores, independentment de si l'usuari ha eliminat o no el compte. I es garanteix el dret de supressió segons l'article 17 del Reglament General de Protecció de dades[16].

3.2. Front-End

L'aplicació generada amb Angular segueix l'arquitectura d'elements i components d'aquest *framework*, no obstant això, s'ha optat per fer ús d'un enfocament modular. Els mòduls definits s'han basat en els descrits en el capítol 2.2.1. Identificació de components. Per a fer més senzilla l'organització i la reutilització del codi, aquests mòduls han patit alguns canvis a l'hora d'agrupar components.

A continuació es detalla l'estructura rellevant de l'aplicació dividida en directoris i fitxers de configuració:

- `src/`: Conté el codi font de l'aplicació.
 - `app/`: Aquest directori conté els distints mòduls dividits en directoris, el mòdul i el component arrel de l'aplicació, i el fitxer de configuració de les rutes.

A continuació es mostra un fragment representatiu de les rutes definides, on s'apliquen distintes guardes per a restringir l'accés.

```

{
    path: 'gestor/comandes',
    component: ComandaListComponent,
    canActivate: [AuthGuard, RolGuard],
    data: { rolAllowed: [ROL.GESTOR, ROL.ADMIN] },
},
{
    path: 'gestor/users',
    component: GestorUsuarisComponent,
    canActivate: [AuthGuard, RolGuard],
    data: { rolAllowed: [ROL.ADMIN] },
},
{
    path: 'profile',
    component: ProfileComponent,
    canActivate: [AuthGuard]
},
{ path: 'login', component: LoginComponent },

```

Figura 45: Configuració de rutes del *front-end*

- `assets/`: Aquest directori conté els recursos estàtics de l'aplicació com les imatges o els estils globals definits
- `environments/`: Conte els fitxers de configuració de l'API per a l'entorn de producció i proves.

```
export const environment = {
  production: true,
  apiUrl: 'https://ec2-15-188-62-23.eu-west-3.compute.amazonaws.com/api/',
};
```

Figura 46: Configuració de l'API en producció del *front-end*

- `manifest.webmanifest`: Fitxer de configuració de l'aplicació web progressiva (PWA).
- `ngsw-config.json`: Fitxer de configuració del *Service Worker* on s'especifiquen les característiques de la PWA, com la configuració per què emmagatzeme les dades obtingudes de l'API en la memòria cau.

Com es mostra en la figura 47, les dades provinents de l'API tindran la següent configuració: s'emmagatzemaran un màxim de 100 recursos, durant una hora com a màxim. L'aplicació esperarà un segon per a obtenir la resposta de la xarxa i s'emprarà l'estratègia *freshness*, és a dir, primer intentarà obtenir resposta de la xarxa, si no està disponible o dona un *timeout*, recuperarà les dades de la memòria cau.

```
"dataGroups": [
  {
    "name": "api",
    "urls": ["/api"],
    "cacheConfig": {
      "maxSize": 100,
      "maxAge": "1h",
      "timeout": "1s",
      "strategy": "freshness"
    }
  }
]
```

Figura 47: Fragment del fitxer `ngsw-config.json`

Els mòduls han quedat organitzats de la següent manera:

AuthModule: Està format per components, guardes, models i serveis.

En aquest model estan els components de login, logout i reset-password. Les guardes auth i rol comproven si l'usuari està autènticat i el rol d'aquest, respectivament, abans d'accedir a les rutes. Els models de dades defineixen `AuthDTO` i `ROL`, on es detallen els distints rols que hi ha. Les funcionalitats dels serveis definits estan relacionats amb l'autenticació de l'usuari i la caducitat del *token*, i l'interceptor de HTTP, que inclou el *token* a la capçalera de les peticions HTTP.

PortalModule: Està format pels components que conformen el portal web públic i serveis. El component portal està format pel component concerts, que mostra una llista de components concert-item. El component productes mostra una llista de components producte-item. El component cistella mostra una llista de components cistella-item. També hi ha els components música i producteDetall i els serveis que realitzen sol·licituds a l'API sense *token* d'autenticació.

ConcertsModule: Està format per components, models, pipes i serveis. Els components que inclou d'este mòdul són gestor-concert-list que està format per components gestor-concert-item i el component gestor-concert-form. El model de dades definit és ConcertDTO i el pipe mesCat transforma el mes en format numèric d'una data en el nom del mes en català. El servei agrupa les funcionalitats per a obtenir les dades de l'API relacionades amb els concerts.

ProducteModule: Està format per components, models i serveis. Els components agrupats en este mòdul són gestor-producte-list, que inclou components gestor-producte-item i el component gestor-producte-form. Els models de dades relacionats amb els productes són ProducteDTO i TIPUS, on es detallen els tipus de producte definits. Els models de dades relacionats amb els detalls dels productes són DetallProducteDTO, MODEL i TALLA, on es detallen els distints models i talles dels productes de tipus tèxtil. Els serveis agrupen les funcionalitats per a obtenir les dades de l'API relacionades amb els productes i detalls dels productes.

ComandaModule: Està format per components, models, pipes i serveis. Els components que inclou d'este mòdul són comanda-list, comanda-detall realitza-comanda. Els models de dades definits són ComandaDTO, ESTAT, on es detallen els distints estats d'una comanda i ElementComandaDTO. El pipe reduceld redueix l'id d'una comanda. Els serveis agrupen les funcionalitats per a obtenir les dades de l'API relacionades amb les comandes i els elements de les comandes.

UsuariModule: Està format per components, models, serveis i validadors. Els components agrupats en este mòdul són el component register, profile que conté un component profile-form, i el component gestor-usuaris que conté un gestor-usuari-form. El model de dades definit és UsuariDTO i el servei defineix les funcions relacionades amb les operacions dels usuaris. El validador comprova si les contrasenyes introduïdes al formulari són iguals.

SharedModule: En aquest mòdul hi ha components, models i serveis compartits que utilitzen els mòduls definits anteriorment. Els components compartits són els de la capçalera, el peu de pàgina i les pàgines d'avís legal i política de privacitat. El model definit són les diferents opcions de visualització de la capçalera. Un dels serveis maneja les opcions de visualització de la capçalera, l'altre gestiona maneja les dades del *local storage* i l'últim la pujada d'imatges a l'aplicació. I el pipe TruncaDesc que redueix el contingut d'una cadena a cent caràcters i afegeix tres punts al final.

A l'annex 2 es mostren distints fragments dels elements que componen l'aplicació, de manera que siguin representatius.

Per a tindre una base sòlida i coherent en el disseny s'ha fet ús majorment de Bootstrap[17] a la part privada de l'aplicació, on es troben els modals i els formularis. No obstant això, s'han utilitzat les variables de SCSS per a definir els colors i les fonts.

A la part pública de la plataforma web, per a donar-li un estil personalitzat seguint els prototips d'alt nivell, s'han dissenyat els fulls d'estil amb Sass[18], aquesta llenguatge permet fer ús de variables, regles imbrincades i funcions, entre altres avantatges front a CSS, com es mostren en la següent figura.

```

@use "sass:color";

$mq-max-mobile-width: 767px;
$mq-min-tablet-width: 768px;
$mq-max-tablet-width: 1023px;
$mq-min-desktop-width: 1024px;
$header-height: 5rem;
$footer-height: 8rem;
$font-title: "Poppins", sans-serif;
$font-text: "Poppins", sans-serif;
$primary: #a82700;
$secondary: #d8947d;
$light: #f4f1f0;
$dark: #000;
$success: #00a827;
$warning: #d78400;
$danger: #d9d9d9;
$info: #44bee3;
$info-secondary: #b6214a;
$white: #fff;
$danger-transparent: rgba($danger, 0.3);
$font-family-sans-serif: $font-text;
$font-weight-light: 400;
$font-weight-normal: 600;
$font-weight-bold: 800;
$font-weight-base: $font-weight-light;
$navbar-dark-color: $white;
$navbar-dark-hover-color: $light;
$navbar-dark-active-color: $white;

@import "src/assets/styles/variables";

.portada {
  position: relative;

  &__imatge {
    left: 0;
    z-index: 1;
    width: 100%;
    height: auto;
  }

  &__indicador {
    display: none;
    z-index: 0;
    position: fixed;
    bottom: 10vh;
    left: 50%;
    transform: translateX(-50%);
    text-align: center;
    color: $light;
    animation: baixa 2s infinite;
  }

  &__fletxa {
    font-size: 2rem;
  }
}

```

Figura 48: Fragments de codi SCSS del *front-end*

Els fulls d'estil s'han dissenyat seguint la metodologia BEM (Block, Element, Modifier)[19], d'aquesta manera facilita l'enteniment i el manteniment del codi. En la figura 49 es mostra un fragment del codi HTML del component *producte-item*. En aquest fragment es defineix un bloc "producte" format per distints elements. S'observa que primer l'element "producte__columna" conté un modificador per indicar que la columna conté una imatge.

```
<div class="producte">
  <div class="producte_fila">
    <div class="producte_columna producte_columna--img">
      
    </div>
    <div class="producte_columna">
      <div class="producte_detalls">
        <h5 class="producte_nom"><a class="producte_link" href="/producte/{{producte.id}}"
          attr.aria-label="Pàgina del producte {{producte.nom}}">{{producte.nom}}</a></h5>
        <p class="producte_descripcio">{{producte.descripcio | truncaDesc }}</p>
        <h5 class="producte_preu ">{{producte.preu_base }}€</h5>
      </div>
    </div>
  </div>
</div>
```

Figura 49: Codi HTML del component producte-item del *front-end*

La guia d'estil emprada, Code Guide[20], permet estructurar els estils de SCSS i el codi HTML d'una manera coherent, flexible, sostenible i consistent.

Els estils reutilitzables en diferents parts del codi s'han definit dins del directori `assets/styles`, en canvi, aquells propis de cada component s'han definit en el fitxer `.scss` d'aquest.

4. Tests

En aquest capítol es descriuen les ferramentes utilitzades durant el procés de desenvolupament per a garantir el correcte funcionament tant de l'aplicació com de l'API REST.

4.1. Back-End

Per a detectar possibles errors i assegurar que totes les sol·licituds de l'API es comporten com s'espera, s'ha optat per fer ús de l'eina Postman[21], ja que és una interfície gràfica intuïtiva per a testar APIs.

Amb aquesta eina s'ha creat una col·lecció on s'han organitzat les sol·licituds de manera ordenada. A la següent figura es mostra un exemple de la sol·licitud GET concerts amb el *token* d'autenticació.

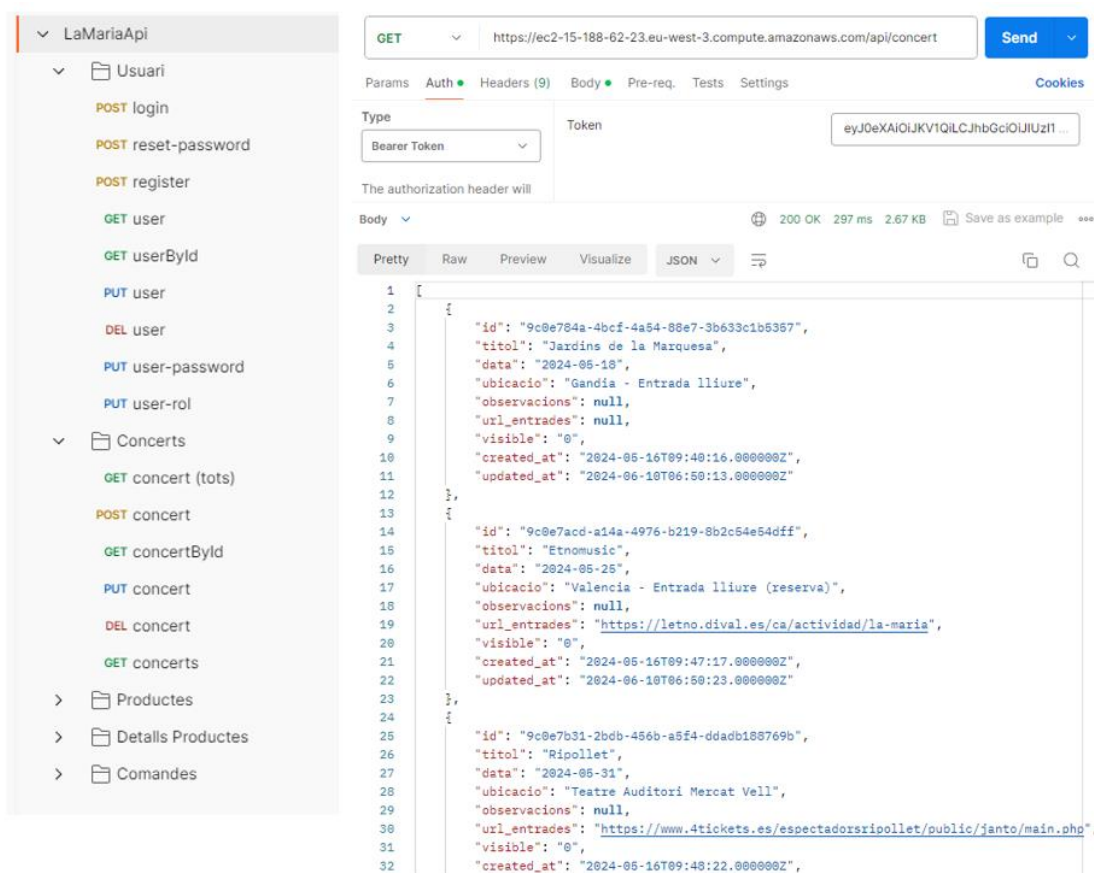


Figura 50: Sol·licitud realitzada en Postman

4.2. Front-End

Per a verificar que el funcionament del codi és correcte, s'han fet ús de les proves unitàries. Aquestes proves permeten testar funcions i mètodes individuals del codi.

En l'aplicació s'ha fet ús del *framework* Jasmine, integrat amb Karma, eina que proporciona Angular. Amb aquesta ferramenta s'han desenvolupat 176 tests unitaris, on es testen els distints mètodes dels components i pipes. D'aquesta manera permet provar com es comporten en distintes condicions i estats.

En executar en la consola **ng test** s'executen les proves definides. Com es mostra a la figura 51, les proves tenen una cobertura sobre la major part del codi. La cobertura *statements* indica el percentatge de codi executat en les proves, la cobertura *branches* mostra el percentatge de condicions del codi testades. La cobertura *funcions* representa el percentatge de funcions provades i per acabar, la cobertura *lines* indica el percentatge de declaracions executades durant les proves.

```

===== Cov
Statements : 83.88% ( 937/1117 )
Branches   : 64% ( 112/175 )
Functions  : 73.38% ( 273/372 )
Lines      : 84% ( 914/1088 )
WARN: 'Navigation triggered outside
  
```

Figura 51: Cobertura proporcionen les proves unitàries

```

Karma v 6.4.3 - connected; test: complete; DEBUG
Jasmine 4.6.0
176 specs, 0 failures, randomized with seed 34241 Finished in 1.333s

GestorProducteFormComponent
  * keysofModel - torna les claus dels Models
  * pujaDetalls - crida a createDetailProducte o actualitzaDetailProducte per a cada detall
  * resetForm - resetja formulari
  * keysofTipus - torna les claus dels tipus
  * pujaImatge - crida a uploadImage si selectedFile no és null
  * actualitzaProducte - crida a updateProducte i pujaImatge
  * actualitzaDetailProducte - crida a updateDetailProducte
  * existeixDetailNoTextil - torna l'index del detall
  * creaProducte - crida a createProducte i pujaImatge
  * creaDetailProducte - crida a createDetailProducte
  * existeixDetailTextil - torna l'index del detall
  * submit - crida a submitFormulari quan producteForm és vàlid
  * keysofTalla - torna les claus de les Talles
  * submitFormulari - estableix isValidFormProducte com true i crida a actualitzaProducte
  * submitFormulari - estableix isValidFormProducte com a true i crida creaProducte
  * submit - estableix isValidFormProducte com a false quan producteForm és invàlid
  * should create
  * tipusFormulari - estableix tipusForm com "edit" quan id no és una cadena buida
  * ngOnInit - should initialize the form with producte data
  * afegexDetail - comprova el tipus de producte i crida existeixDetailNoTextil si tipus no és 1
  * existeixDetailNoTextil - torna l'index -1
  * loadDetailProductes amb èxit des de la subscripció
  * afegexDetail - comprova el tipus de producte i crida existeixDetailTextil si tipus és 1
  * tipusFormulari - estableix tipusForm com "add" quan id és una cadena buida
  * afegexDetail - actualitza la quantitat si el detall ja existeix
  * existeixDetailTextil - torna l'index -1
  * submitFormulari - producte No Textil crida a afegexDetail
  * afegexDetail - afegex un nou detall si no existeix

LogoutComponent
  * should create

ComandaListComponent
  * hasComanda - comprova si el concert conté comandes
  * keysofEstat - torna les claus d'ESTAT
  * loadComandes - carrega les comandes d'un usuari
  * pushConcerts - afegex un concert si conté una comanda
  * pushConcerts - no afegex un concert si no conté una comanda
  * pushConcerts - no afegex un concert si ja es troba en la llista
  * loadConcerts - carrega els concerts
  
```

Figura 52: Informe dels tests unitaris amb Karma

Gràcies a aquests tests, s'han pogut detectar errors en fases primerenques i s'ha comprovat que les funcionalitats principals s'executen com s'espera després de cada canvi. Les mètriques de cobertura indiquen que podrien millorar-se; no obstant això, els resultats són satisfactoris i proporcionen una base sòlida al projecte.

5. Desplegament de l'aplicació

En aquest capítol es durà a terme el desplegament de l'aplicació. En primer lloc, es definiran els requeriments. En segon lloc, es desplegarà l'API REST en local i en el servidor. Per acabar, es desplegarà l'aplicació en local i en el servidor.

5.1. Requeriments

Per a desplegar l'aplicació en un entorn local és necessari comptar amb els següents recursos

Back-End:

- Servidor Apache versió 2.4 o superior
- Base de dades MySQL
- PHP versió 8.2 o superior
- Laravel versió 11.x

Front-End

- NodeJS
- Package manager (npm)
- Angular versió 16

Per a desplegar l'aplicació en el servidor és necessari comptar amb el servidor d'Apache, la base de dades MySQL i amb PHP, ja que tant l'API com el *front-end* estaran prèviament compilats.

5.2. Desplegament *back-end* en local

Per a instal·lar el *back-end* desenvolupat en Laravel s'han de seguir les següents passes:

1. Modificar l'arxiu .env

Afegir els paràmetres necessaris per a la connexió de la base de dades a utilitzar en les següents variables d'entorn:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nom_de_la_bbdd
DB_USERNAME=root
DB_PASSWORD=
```

Figura 53: Configuració local connexió a la base de dades

Per a configurar les rutes on s'emmagatzemaran les imatges al *back-end*, i posteriorment al *front-end*, quan és pugen imatges des de l'API s'han de configurar les variables d'entorn següents

```
INPUT_FILES="C:/xampp/htdocs/la_maria_api/public/public"
OUTPUT_FILES="C:/xampp/htdocs/la-maria-app/src/assets/images/productes"
```

Figura 54: Configuració local rutes emmagatzematge imatges

En cas de voler configurar el sistema de correu electrònic per a la recuperació de contrasenya, s'han de configurar les credencials en les següents variables d'entorn

```
MAIL_MAILER=log
MAIL_HOST=127.0.0.1
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="la_maria@noreply.es"
MAIL_FROM_NAME="${APP_NAME}"
```

Figura 55: Configuració local servidor de correus electrònics

2. Instal·lar les dependències

Per a instal·lar les dependències fent ús del terminal, cal accedir a l'arrel del projecte i executar la instrucció:

```
composer install
```

3. Crear i poblar la base de dades

Per a crear la base de dades i poblar-la amb dades, cal importar el fitxer `la_maria_app.sql` al motor de base de dades MySQL. Aquestes contenen tres usuaris definits, un amb cada rol. Les credencials per a accedir són les següents:

Usuari: admin@uoc.edu

Contrasenya: admin1234

Usuari: gestor@uoc.edu

Contrasenya: gestor1234

Usuari: usuari@uoc.edu

Contrasenya: usuari 1234

Per a migrar i poblar les taules des de Laravel amb dades falses es pot fer executant les següents instruccions:

```
php artisan migrate
php artisan db:seed
```

Aquestes instruccions creen els usuaris definits anteriorment.

4. Iniciar el servidor de Laravel

En executar `php artisan serve` s'alça el servidor de Laravel i ja se'n pot fer ús de l'API

5.3. Desplegament *back-end* en el servidor

Per a instal·lar el *back-end* al servidor s'han de seguir les següents passes:

1. Connectar-se al servidor des de FileZilla mitjançant el protocol SFTP (SSH File Transfer Protocol)

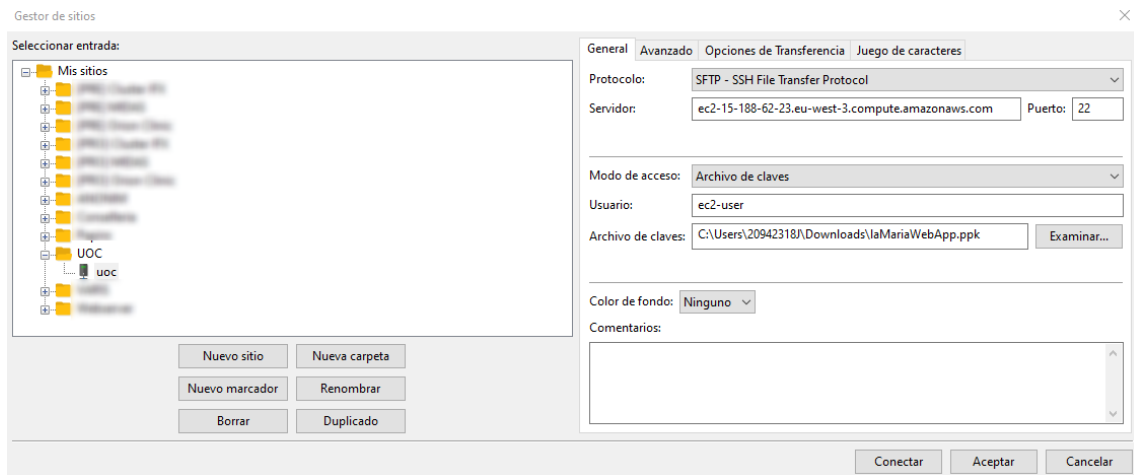


Figura 56: Connexió SFTP al servidor

2. Transferir el projecte al servidor

Copiar el projecte de l'API dins del directori /var/www del servidor

3. Modificar l'arxiu .env

Definir els paràmetres per a indicar que l'aplicació està en producció

```
APP_NAME="La Maria"
APP_ENV=production
APP_KEY=*****
APP_DEBUG=false
APP_TIMEZONE=UTC+1
APP_URL=https://ec2-15-188-62-23.eu-west-3.compute.amazonaws.com
```

Figura 57: Configuració de producció *back-end*

Afegir els paràmetres necessaris per a la connexió de la base de dades a utilitzar en les següents variables d'entorn

```
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=la_maria_app
DB_USERNAME=root
DB_PASSWORD=*****
```

Figura 58: Configuració connexió a la base de dades

Configurar el sistema de correu electrònic per a la recuperació de contrasenya, cal de configurar les credencials en les següents variables d'entorn

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME="webapplamaria@gmail.com"
MAIL_PASSWORD=*****
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="la_maria@noreply.es"
MAIL_FROM_NAME="${APP_NAME}"
```

Figura 59: Configuració servidor de correus electrònics

Configurar la ruta on es rebran i s'emmagatzemaran les imatges pujades a l'aplicació

```
INPUT_FILES="/var/www/la_maria_api/public/public"
OUTPUT_FILES="/var/www/html/la_maria/assets/images/productes"
```

Figura 60: Configuració rutes emmagatzematge imatges

6. Instal·lar les dependències

Cal accedir a l'arrel del projecte i executar la instrucció:

composer install

7. Crear el host virtual per a l'API

Crear el fitxer de configuració al servidor Apache, */etc/httpd/conf.d/api.conf*, per a definir el host virtual de l'API al port 8000.

```
<VirtualHost *:8000>
  ServerName ec2-15-188-62-23.eu-west-3.compute.amazonaws.com:8000/api
  DocumentRoot /var/www/la_maria_api/public

  <Directory /var/www/la_maria_api/public>
    Options FollowSymLinks
    AllowOverride All
    Require all granted

    <LimitExcept GET POST PUT DELETE>
      Require all denied
    </LimitExcept>
  </Directory>

  ErrorLog /var/log/httpd/api-error.log
  CustomLog /var/log/httpd/api-access.log combined
</VirtualHost>
```

Figura 61: Configuració Virtual Host de l'API

Definir el nom del servidor, el directori arrel i un bloc de configuració per al directori /públic.

5.4. Desplegament *front-end* en local

Per a instal·lar el *front-end* desenvolupat en Angular simulant l'entorn de producció per a fer ús del *service worker*, s'han de seguir les següents passes:

1. Instal·lació de dependències

Fent ús del terminal, cal accedir a l'arrel del projecte i executar la instrucció:
npm install

2. Compilar l'aplicació

Per a compilar l'aplicació per a l'entorn de producció s'executa la següent instrucció:

ng build

3. Iniciar el servidor lleuger

Una vegada compilada l'aplicació correctament, per a iniciar el servidor cal accedir al directori on es troben els arxius estàtics amb la instrucció:

```
cd ./dist/la_maria
```

Una vegada en el directori, es pot iniciar el servidor web amb la instrucció:
http-server

En cas de no ser necessari fer ús del *service worker*, des de l'arrel del projecte, es pot iniciar el servidor d'Angular fent ús de la instrucció:
ng serve

5.5. Desplegament *front-end* en el servidor

1. Compilar el projecte

Per a compilar l'aplicació s'executa la següent instrucció:
ng build

2. Transferir el projecte al servidor

Mitjançant FileZilla, copiar el projecte compilat dins del directori `/var/www/html` del servidor

3. Afegir fitxer de configuració `.htaccess` per a gestionar la navegació

Afegir, mitjançant FileZilla, el fitxer `.htaccess` per a gestionar les URL correctament.

```
<IfModule mod_rewrite.c>
  RewriteEngine On

  #Si es per a un fitxer que existeix, seguir
  RewriteCond %{REQUEST_FILENAME} -f [OR]
  RewriteCond %{REQUEST_FILENAME} -d
  RewriteRule ^ - [L]

  #Si no, redirigir a index.html
  RewriteRule ^.*$ /index.html [L]
</IfModule>
```

Figura 62: Fitxer `.htaccess` de l'aplicació

4. Crear el host virtual per a l'aplicació

Crear el fitxer de configuració al servidor Apache, `/etc/httpd/conf.d/app.conf`, per a configurar el host virtual de l'aplicació al port 443 i al port 80.

```
<VirtualHost *:443>
  ServerName ec2-15-188-62-23.eu-west-3.compute.amazonaws.com
  DocumentRoot /var/www/html/la_maria

  <Directory /var/www/html/la_maria>
    Options FollowSymLinks MultiViews
    AllowOverride All
    Require all granted
  </Directory>

  <Location /api>
    ProxyPass http://ec2-15-188-62-23.eu-west-3.compute.amazonaws.com:8000/api
    ProxyPassReverse http://ec2-15-188-62-23.eu-west-3.compute.amazonaws.com:8000/api
  </Location>

  SSLEngine on
  SSLCertificateFile "/etc/httpd/certificate/la-maria.crt"
  SSLCertificateKeyFile "/etc/httpd/certificate/la-maria.key"
  SSLCertificateChainFile "/etc/httpd/certificate/la-maria.pem"
</VirtualHost>

<VirtualHost *:80>
  RewriteEngine On
  RewriteCond %{REQUEST_URI} !^/phpmyadmin [NC]
  RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
</VirtualHost>
```

Figura 63: Configuració Virtual Host de l'aplicació

Per a definir la connexió HTTPS, des del port 443, es defineix el nom del servidor, la configuració del directori.

Per al punt d'accés */api* es defineix un *proxy* que redirigeix la petició al *virtual host* del *back-end*.

S'habilita el protocol SSL i s'especifiquen els arxius dels certificats de seguretat.

Les connexions HTTP, des del port 80, es redirigiran al protocol HTTPS.

6. Instruccions d'ús

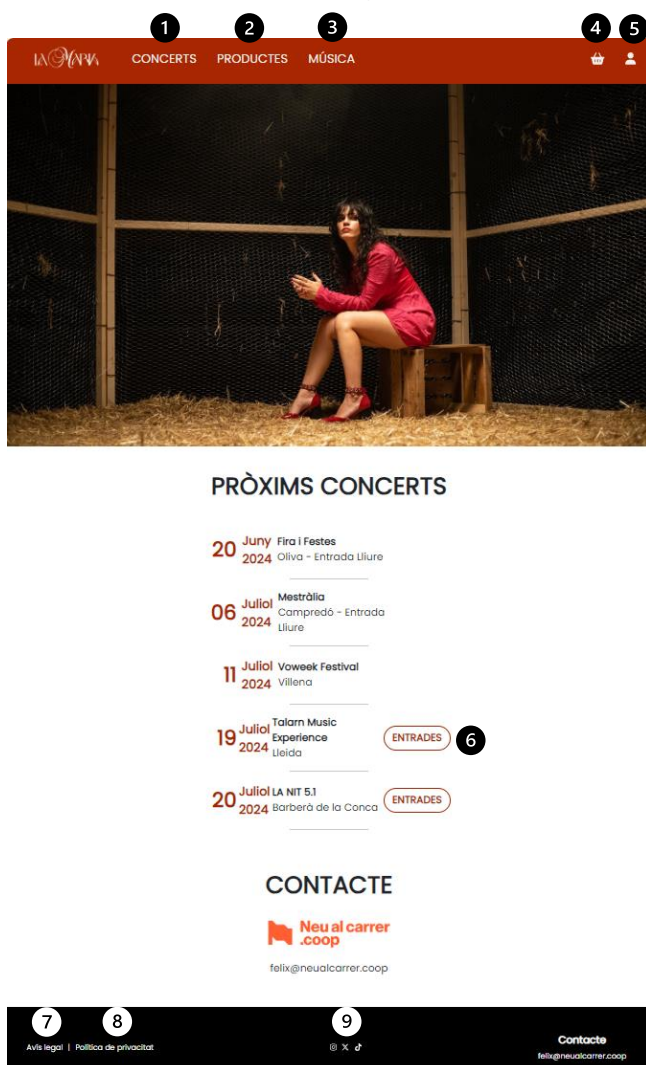
En aquest capítol es detallarà com fer ús de l'aplicació. Aquest capítol es divideix en dos subcapítols, per un costat s'explica el funcionament del portal públic i les accions que es poden realitzar, i en l'altre subcapítol s'explica el funcionament del portal privat.

6.1. Portal públic

6.1.1. Inici

En entrar a l'aplicació la pàgina es divideix en tres parts.

A la part superior de la pàgina hi ha menú principal, amb els punts d'accés a les diferents pàgines. En la part central, es troba la imatge de portada, una llista amb els pròxims sis concerts i la informació de contacte. A la part inferior de la pantalla, hi ha el peu de pàgina amb distints enllaços.



1. Punt d'accés a la pàgina de Concerts
2. Punt d'accés a la pàgina de Productes
3. Punt d'accés a la pàgina de Música
4. Punt d'accés a la pàgina de la Cistella
5. Punt d'accés a la pàgina d'inici de sessió
6. Botó per a accedir a la pàgina de l'entitat encarregada de la venda d'entrades. Aquestes pàgines tenen la seva pròpia mecànica d'ús.
7. Punt d'accés a l'avís legal de la pàgina
8. Punt d'accés a les polítiques de privacitat de la plataforma
9. Enllaços a les xarxes socials de l'artista

Figura 64: Pàgina inicial de l'aplicació

En dispositius mòbils i tauletes el menú principal es comprimeix. En polsar damunt de **MENU** es desplega a l'esquerra de la pàgina. Oferint les mateixes opcions que al menú principal.



Figura 65: Menú principal versió mòbil

6.1.2. Consultar els pròxims concerts

En aquesta pàgina es troben tots els concerts que estan per vendre.

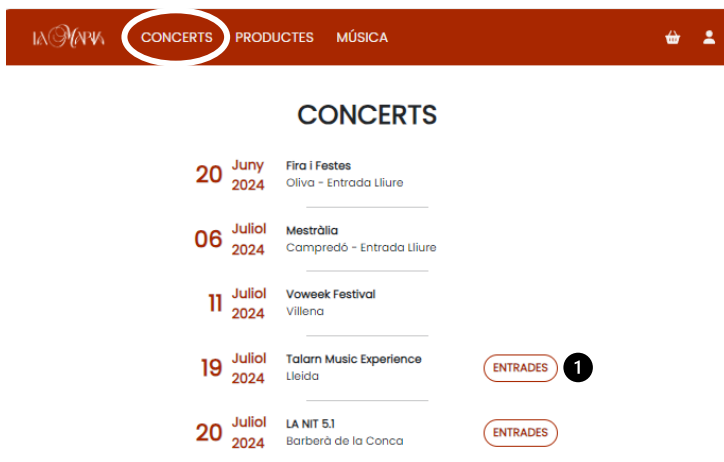


Figura 66: Pàgina de concerts

1. Botó per a accedir a la pàgina de l'entitat encarregada de la venda d'entrades. Aquestes pàgines tenen la seua pròpia mecànica d'ús.

6.1.3. Consulta els productes disponibles

En aquesta pàgina es troben els productes disponibles.

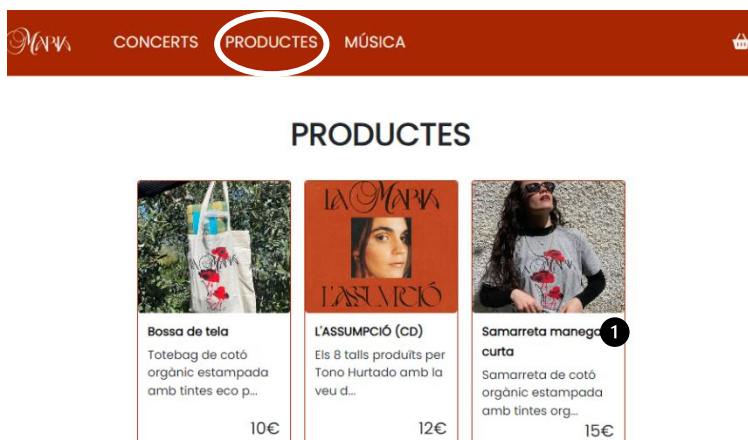


Figura 67: Pàgina productes

1. Accedeix als detalls del producte.

Des de la pàgina de detall de producte es pot:



Figura 68: Pàgina detall producte

1. Escollir la quantitat de productes
2. Escollir el model del producte
3. Escollir la talla del producte
4. Afegir a la cistella.
5. Tornar a la pàgina anterior

6.1.4. Música

En aquesta pàgina es poden reproduir els treballs de l'artista sense haver d'eixir del portal.

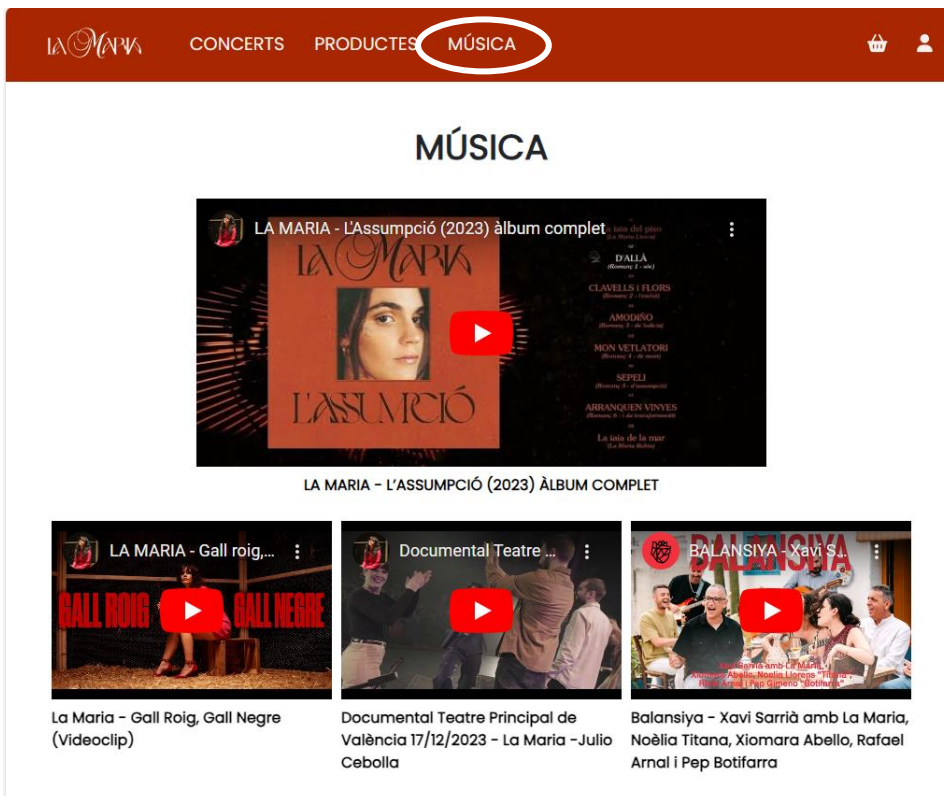


Figura 69: Pàgina música

6.1.5. Cistella

Des de la cistella es poden realitzar les següents accions:

LA MEUA CISTELLA

	1 X	Samarreta manega curta Entallada-S	15 €		1
	1 X	L'ASSUMPCIÓ (CD)	12 €		

2 Preu total (amb IVA): 32.67€

3 CANCEL·LAR 4 CONTINUAR

El cercle indica que hi ha productes dins la cistella

1. Eliminar un producte
2. Consultar el preu total amb l'IVA inclòs.
3. Navegar cap arrere. Si es polsa aquest botó no s'esborren els productes
4. Continuar la reserva

Figura 70: Pàgina cistella

En polsar el botó “Continuar”, la pàgina es redirigirà a la pàgina de reserva (vegeu 6.2.1). Per a accedir a aquesta pàgina si s'ha iniciat sessió mostrarà la pàgina d'inici de sessió (vegeu 6.1.6) abans de continuar.

6.1.6. Iniciar sessió

INICIAR SESSIÓ

EMAIL: 1

CONTRASENYA: 2

He oblidat la contrasenya 4

3 **ENTRA**

No tens compte? [Registra't!](#) 5

1. Inserir un correu electrònic
2. Inserir una contrasenya
3. Validació de les credencials
4. Punt d'accés a la pàgina de recuperació de la contrasenya.
5. Punt d'accés a la pàgina de registre

Figura 71: Pàgina inici de sessió

Si les credencials són correctes la pàgina es redirigirà a la pàgina a la qual s'intentava accedir. En cas de no iniciar la sessió amb èxit es mostrarà una alerta en roig indicant l'error.

6.1.7. Registre

A la pàgina de registre s'accedeix des de la pàgina d'inici de sessió.

[He oblidat la contrasenya](#)

ENTRA

No tens compte? [Registra't!](#)

Figura 72: Punt d'accés de registre

CREAR COMPTE

NOM:

COGNOMS:

TELÈFON:

EMAIL:

CONTRASENYA:

CONFIRMA CONTRASENYA:

REGISTRAT

1. Introduir el nom
2. Introduir els cognoms
3. Introduir número de telèfon
4. Introduir un correu electrònic
5. Introduir la contrasenya.
6. Confirmar la contrasenya. S'haurà d'introduir de nou la mateixa contrasenya, si no coincideixen el sistema mostrarà un error indicant-ho.
7. Confirmar el registre. El sistema indicarà amb una alerta si s'ha realitzat satisfactòriament l'acció o no.
8. Punt d'accés a la pàgina d'inici de sessió

Ja tens compte? [Inicia Sessió!](#)

Figura 73: Pàgina de registre

6.1.8. Recuperar la contrasenya

A la pàgina de recuperar la contrasenya s'accedeix des de la pàgina d'inici de sessió

CONTRASENYA:

[He oblidat la contrasenya](#)

ENTRA

Figura 74: Punt d'accés a recuperar contrasenya

Per a recuperar la contrasenya cal:

RECUPERAR LA CONTRASENYA

EMAIL:

RESETEJAR CONTRASENYA

1. Inserir el correu electrònic amb el qual s'ha registrat el compte.
2. Confirmar l'acció

El sistema mostrarà un missatge indicant que si el correu electrònic introduït està registrat, rebrà un missatge amb una nova contrasenya que podrà canviar des del seu compte.

Figura 75: Pàgina recuperar contrasenya

6.2. Portal privat

Per a accedir a les pàgines del portal privat l'usuari o usuària ha d'haver-se autenticat prèviament des de la pàgina d'inici de sessió.

En autenticar-se s'habilitaran les opcions del menú secundari, on, segons el rol assignat al perfil, es mostraran unes opcions o altres.

GESTIÓ PRODUCTES	1	1. Punt d'accés a la pàgina Gestió de productes. Accediran els usuaris amb rol de gestió.
GESTIÓ CONCERTS	2	2. Punt d'accés a la pàgina Gestió de concerts. Accediran els usuaris amb rol de gestió.
GESTIÓ COMANDES	3	3. Punt d'accés a la pàgina Gestió de comandes. Accediran els usuaris amb rol de gestió.
<hr/>		
GESTIÓ USUARIS/ES	4	4. Punt d'accés a la pàgina Gestió d'usuaris/es. Accediran els usuaris amb rol d'administrador.
	5	5. Punt d'accés a la pàgina Les meues reserves. Accediran tots els usuaris.
LES MEUES RESERVES	5	
EL MEU COMPTE	6	6. Punt d'accés a la pàgina El meu compte. Accediran tots els usuaris.

Figura 76: Menú secundari

6.2.1. Reservar la comanda

A la pàgina de reserva dels productes s'accedeix des de la cistella, no obstant això, s'ha de tindre la sessió iniciada.

LA MEUA CISTELLA

1 X	Samarreta manega curta Entallada-S	15 €	
1 X	L'ASSUMPCIÓ (CD)	12 €	

u total (amb IVA): **32.67€**

Figura 77: Punt d'accés a la pàgina reservar

Des d'aquesta pàgina es pot acabar la comanda

RESERVA

Tria el concert d'entrega **1**

20/06/2024 - Fira i Festes

06/07/2024 - Mestràlia

11/07/2024 - Voweeek Festival

19/07/2024 - Talarn Music Experience

20/07/2024 - LA NIT 5.1

Admin

COGNOMS: **3**

Usuari de prova

EMAIL:

admin@uoc.edu

TELÈFON:

600000000

OBSERVACIONS: **2**

4 CANCEL·LAR
5 CONTINUAR

1. Escollir el concert on s'entregarà i pagarà la comanda
2. Inserir observacions per a l'usuari gestor.
3. Consultar les dades de contacte que es proporcionarà a l'organització. En cas de voler canviar les dades de contacte, s'han de modificar des de la pàgina del perfil de l'usuari.
4. Cancel·lar l'acció i tornar a la pàgina de la cistella
5. Confirmar la reserva. Aquesta acció redirigeix a la pàgina de les meues reserves.

Figura 78: Pàgina reservar

6.2.2. Les meues reserves

En aquesta pàgina es mostren les comandes realitzades, per l'usuari o usuària autenticada, dividides per concerts.

IA3/WA
CONCERTS PRODUCTES MÚSICA

GESTIÓ PRODUCTES
GESTIÓ CONCERTS
GESTIÓ COMANDES
GESTIÓ USUARIS/ES
LES MEUES RESERVES
EL MEU COMPTE

COMANDES

▼ Estat: **1** Tots ▼ Concerts: **2** Tots ▼

Voweeek Festival

nº Com.	Estat	Data	Preu
3 562d88f4e55c	Cancel·lada	02/06/2024	24.20 €

LA NIT 5.1

nº Com.	Estat	Data	Preu
9398f1fb57e0	Pendent	10/06/2024	32.67 €

1. Filtrar les comandes per estat
2. Filtrar les comandes per concert
3. Accedir al detall de la comanda

Figura 79: Pàgina Les meues reserves

En aquesta pàgina es pot consultar el detall de la comanda.

1 ← COMANDA 0398ffb5f7e0

Estat: **2**
Pendent

Concert LA NIT 5.1

Producte	Model	Talla	Unitats	Preu
Samarreta manega curta	Entallada	S	1	18.15 €
L'ASSUMPCIÓ (CD)			1	14.52 €
TOTAL			2	32.67 €

Dades de contacte **3**

NOM: Admin
 COGNOMS: Usuari de prova
 EMAIL: admin@uoc.edu
 TELEFON: 600000000

OBSERVACIONS:

1. Tornar a la pàgina anterior
2. Modificar l'estat de la comanda. Acció habilitada si el rol és gestor o administrador.
3. Dades de contacte de la persona que ha realitzat la comanda. Vista habilitada si el rol és gestor o administrador.

Figura 80: Pàgina de detall d'una comanda

6.2.3. El meu compte

En aquesta pàgina es poden consultar i modificar les dades de l'usuari/ària autenticada.

LA OPA

CONCERTS
 PRODUCTES
 MÚSICA
 GESTIÓ PRODUCTES
 GESTIÓ CONCERTS
 GESTIÓ COMANDES
 GESTIÓ USUARIS/ES

EL MEU COMPTE

Admin Usuari de prova
 admin@uoc.edu

1 Dades personals
2 Canvia la contrasenya
3 Elimina el compte

CISTELLA
 LES MEUES RESERVES
EL MEU COMPTE
 TANCAR SESSIÓ

1. Mostra el formulari per a modificar les dades personals.
2. Mostra el formulari per a modificar la contrasenya
3. Mostra el formulari per a eliminar el compte

Figura 81: Pàgina El meu compte

- a. Formulari per a modificar les dades personals
 1. Modificar nom
 2. Modificar cognoms
 3. Modificar telèfon
 4. Modificar correu electrònic
 5. Tancar el formulari
 6. Desar els canvis. El sistema mostrarà una alerta indicant si s'ha realitzat l'acció satisfactòriament o si ha hagut algun problema abans de tancar el formulari

Figura 82: Formulari modificar dades personals

- b. Formulari per a modificar contrasenya
 1. Inserir contrasenya actual
 2. Inserir nova contrasenya
 3. Confirmar la contrasenya. S'haurà d'introduir de nou la mateixa contrasenya, si no coincideixen el sistema mostrarà un error indicant-ho
 4. Tancar el formulari
 5. Desar els canvis. El sistema mostrarà una alerta indicant si s'ha realitzat l'acció satisfactòriament o si ha hagut algun problema abans de tancar el formulari

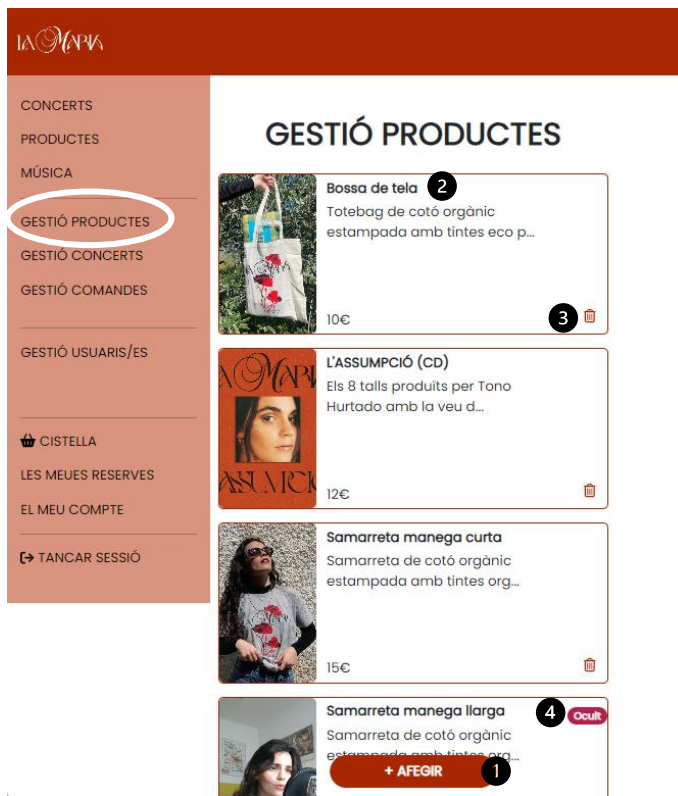
Figura 83: Formulari modificar contrasenya

- c. Formulari per a eliminar compte
 1. Tancar el formulari
 2. Eliminar compte. El sistema mostrarà una alerta indicant si s'ha realitzat l'acció satisfactòriament abans de tancar la sessió

Figura 84: Formular eliminar compte

6.2.4. Gestió dels productes

En aquesta pàgina es gestionen tots els productes de l'aplicació. Des d'aquesta pàgina es poden realitzar les següents accions:



1. Crear un nou producte. Obri un formulari on inserir les dades
2. Modificar un producte existent. Obri un formulari on modificar les dades
3. Eliminar un producte existent. El producte s'eliminarà sempre que no estiga associat a cap comanda
4. Indica que no és visible al portal públic.

Figura 85: Pàgina gestió de productes

- a. Formulari per a crear i modificar un producte
 2. Afegir o modificar el nom del producte
 3. Afegir o modificar el preu
 4. Afegir o modificar l'IVA
 5. Afegir o modificar la descripció
 6. Assignar un tipus al producte
 7. Afegir o modificar la quantitat de estoc
 8. Afegir o modificar la imatge del producte
 9. Mostrar o ocultar el producte
 10. Tancar el formulari
 11. Desar els canvis. El sistema mostrarà una alerta indicant si s'ha realitzat l'acció satisfactòriament o si ha hagut algun problema abans de tancar el formulari.

Figura 86: Formulari afegir o modificar un producte

6.2.5. Gestió dels concerts

En aquesta pàgina es gestionen tots els concerts de l'aplicació. Des d'aquesta pàgina es poden realitzar les següents accions:

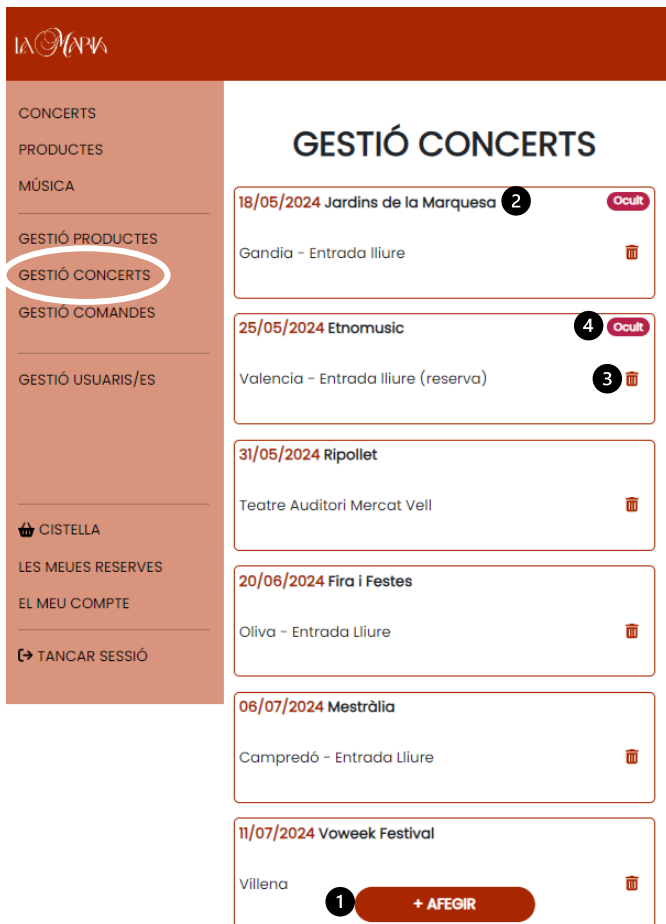


Figura 87: Pàgina gestió de concerts

1. Afegir o modificar el títol del concert
2. Afegir o modificar la ubicació
3. Afegir o modificar l'URL de les entrades
4. Afegir o modificar observacions
5. Afegir o modificar la data del concert
6. Mostrar o ocultar el concert
7. Tancar el formulari
8. Desar els canvis. El sistema mostrarà una alerta indicant si s'ha realitzat l'acció satisfactòriament o si ha hagut algun problema abans de tancar el formulari.

1. Crear un nou concert. Obri un formulari on inserir les dades
2. Modificar un concert existent. Obri un formulari on modificar les dades
3. Eliminar un concert existent. El concert s'eliminarà sempre que no estiga associat a cap comanda
4. Indica que no és visible al portal públic.

- a. Formulari per a crear i modificar un concert

Figura 88: Formulari afegir o modificar un concert

6.2.6. Gestió Comandes

En aquesta pàgina es mostren totes les comandes realitzades dividides per concerts.

nº Com.	Estat	Data	Preu
1fe4042fa34a	Recollida	16/05/2024	24.10 €
Mestràlia			
nº Com.	Estat	Data	Preu
36d908acca32d	Pendent	06/06/2024	18.15 €
Voweeek Festival			
nº Com.	Estat	Data	Preu
562a864de55c	Cancel·lada	02/06/2024	24.20 €
3ac12fa3be4b	Pendent	04/06/2024	32.67 €
LA NIT 5.1			
nº Com.	Estat	Data	Preu
0398fb5f7e0	Pendent	10/06/2024	32.67 €

1. Filtrar les comandes per estat
2. Filtrar les comandes per concert
3. Accedir al detall de la comanda. (Veure punt 5.2.2)

Figura 89: Pàgina gestió de comandes

6.2.7. Gestió Usuaris/es

En aquesta pàgina es gestionen els rols dels usuaris i usuàries registrats a l'aplicació. Des d'aquesta pàgina es poden realitzar les següents accions.

Rols	Cerca	Usuari	Rol
Tots		Admin Usuari de prova	ADMIN
		admin@uoc.edu	
		Gestor Usuari de prova	GESTOR
		gestor@uoc.edu	
		Usuari Usuari de prova	USER
		usuari@uoc.edu	

1. Filtra pel rol dels usuaris.
2. Filtra per una cadena de caràcters. Es mostren les coincidències amb els noms, cognoms i correus electrònics.
 - a. Modificar el rol d'un usuari. Obri un formulari on modificar el camp.

Figura 90: Pàgina gestió d'usuaris/àries

- a. Formulari per a crear i modificar un concert
 1. Modifica el rol
 2. Tancar el formulari
 3. Desar els canvis. El sistema mostrarà una alerta indicant si s'ha realitzat l'acció satisfactòriament o si ha hagut algun problema abans de tancar el formulari.

MODIFICA ROL

USUARI/ÀRIA:
Usuari Usuari de prova

EMAIL
usuari@uoc.edu

ROL: 1
USER

2 CANCEL·LAR 3 DESAR

Figura 91: Formulari modifica rol

7. Bugs

Com a totes les aplicacions web s'ha procurat que siga un sistema robust, complet i sense errors. No obstant això, s'ha detectat que, en pujar la imatge d'un producte, malgrat que es puja correctament al servidor, no refresca la vista, per tant, fa la sensació que hi ha hagut un error i no s'ha pujat correctament. Aquest error, per falta de temps, es corregirà en una futura versió.

8. Conclusions i treballs futurs

Aquest projecte s'ha dut a terme fent ús dels coneixements adquirits al llarg de les assignatures cursades del màster, permetent així posar en pràctica les ferramentes, tècniques, processos i tecnologies estudiades.

Els objectius principals d'aquest projecte era dissenyar i implementar un portal web junt amb el seu gestor de contingut, adaptable als diferents tipus de dispositius, per a l'artista valenciana, La Maria. A més, havia de desenvolupar-se un sistema de permisos basat en rols, complir les regles d'accessibilitat i ser una interfície senzilla d'utilitzar. Es pot dir, que s'han complert tots els objectius marcats a l'inici del projecte. No obstant això, en un principi es va plantejar un projecte més extens, però en tindre un nombre limitat d'hores es va decidir reduir-ne el contingut i adaptar la quantitat de funcionalitats al temps disponible. Per un altre costat, la primera planificació que es va realitzar era poc realista amb les hores dedicades al desenvolupament de l'aplicació, es va decidir ampliar-les i gràcies a aquesta modificació s'ha complert la planificació d'una manera sorprenentment ajustada.

A mesura que ha anat avançant el projecte, ha sigut necessari realitzar canvis com l'eliminació de les dades personals d'un usuari, en lloc de la completa eliminació d'aquest, per a complir amb la normativa i no tindre conflictes amb les claus externes de la base de dades. Un altre canvi realitzat és la reorganització dels mòduls plantejats en el capítol 2.2.1. Identificació de components a l'hora de desenvolupar el *front-end* per fer més senzilla l'organització i la reutilització del codi. I, en últim lloc, s'ha afegit a la pantalla inicial la informació de contacte baix dels concerts, per tal de dotar de més contingut a la pàgina principal.

Si s'analitzen els impactes previstos al capítol 1.3. Impacte en sostenibilitat, ètic-social i de diversitat, es pot confirmar que s'ha aconseguit desenvolupar una solució que els complisca. En matèria de sostenibilitat s'ha aconseguit publicar els concerts via web i es poden reservar productes des de la plataforma, d'aquesta manera la solució, segons l'anàlisi feta en el capítol 1.3., promou l'ODS 7 – Affordable and clean energy, l'ODS 12 – Responsible consumption and production i l'ODS 13 – Climate action. En els aspectes ètics i de responsabilitat social, la solució desenvolupada recull el mínim de dades personals possibles i s'han detallat les polítiques de privacitat i l'avís legal sobre el tractament d'aquestes emmarcades en la legislació vigent. D'aquesta manera la solució, segons l'anàlisi feta en el capítol 1.3., promou l'ODS 8 – Decent work and economic growth. Per acabar, en termes d'igualtat i diversitat, s'ha desenvolupat la solució complint en la mesura del possible la norma WCAG2. Quant al llenguatge inclusiu s'ha procurat crear un entorn equitatiu. Quan es parla d'usuaris i usuàries com a compte registrat, no existeix una paraula que prescindisca de la descripció de gènere, per tant, s'ha optat per utilitzar la nomenclatura usuaris/es. D'aquesta manera la solució, segons l'anàlisi feta en el capítol 1.3., promou l'ODS 5 – Gender equality i l'ODS 10 – Reduced inequalities.

Com s'ha comentat, en un principi es va plantejar un projecte més extens, dins d'aquestes funcionalitats que es van retirar hi havia les següents idees:

- Mòdul de notícies, on des del gestor de contingut, es pogueren introduir els enllaços de les notícies relacionades amb l'artista i, igual que amb els productes i els concerts, es mostraren al portal públic.
- Mòdul de música, on des del gestor de contingut, gestionar els vídeos que es mostren a la pàgina "Música" i els diferents enllaços a les xarxes socials de música.
- Un formulari de contacte, on poder contactar amb l'artista d'una manera centralitzada i oficial.
- Mòdul d'estil, on des del gestor poder canviar el color primari, secundari i el tipus de lletra. Aquest mòdul haurà d'analitzar-lo amb detall perquè el resultat no permeta que s'incomplisquen les normes d'accessibilitat WCGA2.

Aquest projecte s'ha desenvolupat amb la previsió de publicar-se en un futur, per tant, es realitzaran les millores descrites i es corregiran els errors detectats. A més es posarà en producció, iniciant així la fase de manteniment, on es corregiran possibles errors que es puguin detectar en l'ús real de l'aplicació.

9. Glossari

Codificació en Base64: Mètode de codificació que converteix grups de 3 bytes en seqüències de 4 caràcters. L'alfabet Base64 està format per 64 caràcters, majúscules, minúscules números i els caràcters especials '+' i '/'

Mobile first: Mètode de disseny que enfoca, des de l'inici, el desenvolupament d'un lloc web a dispositius mòbils. Posteriorment el disseny s'adapta a dispositius més grans com tauletes o pantalles d'escriptori.

Norma WCAG2: Normativa d'accessibilitat web que defineix les pautes a seguir per a crear contingut web accessible per a persones amb discapacitats.

Service Worker: Scripts de JavaScript que fa d'intermediaris entre la xarxa i el navegador i s'executen en segon plànol.

Objectius de Desenvolupament Sostenible (ODS): Objectius creats per les Nacions Unides per a promoure el desenvolupament sostenible, posant el focus en la igualtat i dignitat de les persones i el respecte del medi ambient.

Protocol HTTPS: Protocol a nivell d'aplicació, basat en el protocol HTTP, destinat a proporcionar una transferència segura de dades d'hipertext.

Protocol SFTP (SSH File Transfer Protocol): Protocol per a la transferència i manipulació de fitxers a nivell d'aplicació en un flux de dades segur.

Protocol SSL (Secure Socket Layer): Protocol a nivell de transport que permet xifrar el tràfic de dades entre el navegador i el servidor web

Proxy: Intermediari que reenvia les sol·licituds realitzades pel client al servidor de back-end corresponent.

Timeout: Mesura de control de temps emprada per a limitar el temps d'espera entre operacions o processos.

Virtual Host: Configuració que permet allotjar múltiples llocs web en un únic servidor físic. Cada un té el seu propi domini i configuració independent.

10. Bibliografia

1. Prats i Camps, J. [Jesús]. (2023). *La Maria* [fotografia].
<https://www.vilaweb.cat/noticies/la-maria-disc-nou-assumpcio/>
2. Jefatura del Estado. *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*, Boletín Oficial del Estado n.º 294 (2018).
3. JWT.io. (2015). *Introduction to JSON Web Tokens*.
<https://jwt.io/introduction>
4. Angular. (2023). *HTTP - Intercept requests and responses*.
<https://angular.io/guide/http-intercept-requests-and-responses>
5. Angular. (2023). *Preventing unauthorized access*
<https://angular.io/guide/router#preventing-unauthorized-access>
6. W3C. (2024). *WCAG 2.2 Understanding Docs – Contrast(Enhanced) (Level AAA)*.
<https://www.w3.org/WAI/WCAG22/Understanding/contrast-enhanced>
7. Fonticons, Inc. (2024). *Font Awesome*.
<https://fontawesome.com/>
8. Fadra Estudi. (2023). *Gall roig, gall negre de La Maria* [fotografia]
<https://www.viasona.cat/noticia/4400/la-maria-reversiona-la-mitica-peca-gall-roig-gall-negre>
9. Neu al Carrer Coop. (2021). *imatge genèrica xarxes* [fotografia]
<https://www.neualcarrer.coop/images/imatge-generica-xarxes.png>
10. Riera Boronat, A. [Alexandra] i Cañamas Carretero, I. [Irene]. (2023). *Marxandatge de La Maria*.
https://www.instagram.com/p/Cqi8byroOw8/?img_index=1
11. Branca Studio. (2023). *L'Assumpció* [fotografia]
https://www.instagram.com/lamariabs_/p/CnWe3HPsi8q/?img_index=1
12. Bertomeu Soria, M. [Maria]. [La Maria]. (2023, 24 febrer). *LA MARIA - L'Assumpció (2023) àlbum complet* [vídeo en línea]. Youtube.
https://www.youtube.com/watch?v=vC6Eg14yhng&ab_channel=LaMaria
13. Bertomeu Soria, M. [Maria]. [La Maria]. (2023, 28 setembre). *LA MARIA – Gall roig, gall negre (videoclip)* [vídeo en línea]. Youtube.
https://www.youtube.com/watch?v=0aj3RW7aC38&ab_channel=L aMaria
14. Bertomeu Soria, M. [Maria]. [La Maria]. (2024, 28 gener). *Documental Teatre Principal de València 17/12/2023 - La Maria - Julio Cebolla* [vídeo en línea]. Youtube.
https://www.youtube.com/watch?v=b0QcoY3Amyc&ab_channel=L aMaria
15. Sarrià i Batlle, X. [Xavier]. [Xavi Sarrià]. (2023, 4 octubre). *BALANSIYA - Xavi Sarrià amb La Maria, Noèlia Titana, Xiomara Abello, Rafel Arnal i Pep Botifarra* [vídeo en línea]. Youtube.
https://www.youtube.com/watch?v=iCUACBxDJOg&ab_channel=XaviSarri%C3%A0
16. Agencia Española de Protección de Datos. (2024, 8 març). *Derecho de supresión ("al olvido")*.

- <https://www.aepd.es/derechos-y-deberes/conoce-tus-derechos/derecho-de-supresion-al-olvido>
17. Bootstrap. (2023). *Bootstrap · The most popular HTML, CSS, and JS library in the world*. .
<https://getbootstrap.com>
18. Sass (2024). *Sass: Documentation*
<https://sass-lang.com/documentation/>
19. BEM (2024). *BEM — Block Element Modifier*.
<https://getbem.com>
20. @mdo (2023). *Code Guide Standards for developing consistent, flexible, and sustainable HTML and CSS*.
<https://codeguide.co/>
21. Postman, Inc. (2024). *API Tools | Postman API Platform*
<https://www.postman.com/product/tools/>

Annex 1. Extractes de codi font *back-end*

El present annex mostra alguns dels extractes de codi més rellevant respecte a la codificació de l'API de La Maria WebApp.

El següent fragment, mostra la lògica emprada per a iniciar la sessió, on es comproven les credencials i es genera el *token* d'autenticació.

```
/**
 * Comprovar credencials i generar token
 */
public function login(Request $request){
    $authUsuari = new Auth;
    $authUsuari->email = $request->email;
    $authUsuari->password = $request->password;
    $usuari = Auth::where('email', '=', $authUsuari->email)->first();
    if (!$usuari) {
        return response()->json(['message' => "L'usuari no existeix"], 404);
    }
    if (Hash::check($authUsuari->password, $usuari->password)) {
        $token = JWTAuth::fromUser($usuari);
        $usuari->token = $token;
        return response()->json($usuari,200);
    }
    return response()->json(['message'=>'Contrasenya incorrecta'],400);
}
```

El següent fragment, mostra la lògica emprada en el controlador ConcertController de l'API a l'hora de realitzar una consulta a la base de dades, inserir noves dades, actualitzar les existents i eliminar-ne.

```
class ConcertController extends Controller {

    /**
     * Obtenir concerts visibles
     */
    public function next() {
        $concerts = Concert::where('visible', '=', 1)->orderBy('data')->get();
        return response()->json($concerts,200);
    }

    /**
     * Obtenir concerts ordenats per la data
     */
    public function index() {
        $concerts = Concert::orderBy('data')->get();
        return response()->json($concerts,200);
    }
}
```

```

/**
 * Crear un concert
 */
public function store(Request $request) {
    $concert = new Concert;
    $concert->titol = $request->titol;
    $concert->data = $request->data;
    $concert->ubicacio = $request->ubicacio;
    $concert->observacions = $request->observacions;
    $concert->url_entrades = $request->url_entrades;
    $concert->visible = $request->visible;
    $concert->save();
    return response()->json($concert,201);
}

/**
 * Obtenir un concert per id
 */
public function show(string $id) {
    $concert = Concert::find($id);
    return response()->json($concert,200);
}

/**
 * Actualitzar un concert
 */
public function update(Request $request, string $id) {
    $concert = Concert::findOrFail($id);
    $concert->update($request->all());
    return response()->json($concert,200);
}

/**
 * Eliminar un concert
 */
public function destroy(string $id) {
    $concert = Concert::findOrFail($id)->delete();
    return response()->json($concert,200);
}
}

```


Annex 2. Extractes de codi font *front-end*

El present annex mostra alguns dels extractes de codi representatius de la codificació de La Maria WebApp. Els fragments de codi seleccionats són representatius de cada guarda, model, servei, component, pipe i validador.

El següent fragment mostra la lògica emprada en la guarda AuthGuard. Aquesta guarda s'executa quan s'intenta accedir a una ruta protegida. El codi emmagatzema la ruta a la qual es vol accedir, comprova si l'usuari està autenticat des del servei AuthService. Si ho està permetrà accedir a la ruta, si no ho està, redirigeix a l'usuari o usuària a la pàgina d'inici de sessió.

```
export const AuthGuard: CanActivateFn = (
  route: ActivatedRouteSnapshot,
  state: RouterStateSnapshot
):
  | Observable<boolean | UrlTree>
  | Promise<boolean | UrlTree>
  | boolean
  | UrlTree => {
  const authService = inject(AuthService);
  const router = inject(Router);

  authService.setUrl(state.url);

  return authService.usuariAutenticat()
    ? true
    : router.createUrlTree(['/login']);
};
```

El següent fragment de codi mostra el model ComandaDTO emprat dins del mòdul comanda. Aquesta classe, formada per distints paràmetres, proporciona un constructor per a inicialitzar una comanda dins dels components.

```
export class ComandaDTO {
  id: string;
  data: Date;
  observacions: string;
  preu_total: number;
  estat: number;
  usuari_id: string;
  concert_id: string;

  constructor(
    id: string,
    data: Date,
    observacions: string,
    preu_total: number,
    estat: number,
    usuari_id: string,
    concert_id: string
  ) {
    this.id = id;
    this.data = data;
    this.observacions = observacions;
  }
};
```

```

    this.preu_total = preu_total;
    this.estat = estat;
    this.usuari_id = usuari_id;
    this.concert_id = concert_id;
  }
}

```

Aquest fragment de codi mostra el servei definit al mòdul concert. Aquesta classe, formada per distints mètodes, és l'encarregada de gestionar les operacions a través de sol·licituds HTTP. Utilitza `HttpClient` per a interactuar amb l'API, per a obtenir, modificar i eliminar les dades, i `BehaviorSubject` per a permetre la reactivitat en les parts de l'aplicació on es mostren aquestes dades.

```

export class ConcertService {
  private urlApi: string;

  private concertSeleccionatSubject = new BehaviorSubject<any>(null);
  concertSeleccionat$: Observable<ConcertDTO> =
    this.concertSeleccionatSubject.asObservable();

  private concertsSubject = new BehaviorSubject<ConcertDTO[]>([]);
  concerts$ = this.concertsSubject.asObservable();

  constructor(private http: HttpClient) {
    this.urlApi = `${environment.apiUrl}concert`;
  }

  getConcerts(): void {
    this.http
      .get<ConcertDTO[]>(this.urlApi)
      .subscribe((concerts) => this.concertsSubject.next(concerts));
  }

  createConcert(concert: ConcertDTO): Observable<ConcertDTO> {
    return this.http.post<ConcertDTO>(this.urlApi, concert).pipe(
      tap((newConcert) => {
        this.concertsSubject.next([...this.concertsSubject.value,
          newConcert]);
      })
    );
  }

  getConcertById(concertId: string): Observable<ConcertDTO> {
    return this.http.get<ConcertDTO>(this.urlApi + '/' + concertId);
  }

  updateConcert(
    concertId: string,
    concert: ConcertDTO
  ): Observable<ConcertDTO> {
    return this.http
      .put<ConcertDTO>(this.urlApi + '/' + concertId, concert)
      .pipe(
        tap((updatedConcert) => {

```

```

        const currentConcerts =
        this.concertsSubject.value.map((concert) =>
        concert.id === updatedConcert.id ? updatedConcert : concert
        );
        this.concertsSubject.next(currentConcerts);
    })
  );
}

deleteConcert(concertId: string): Observable<any> {
  return this.http.delete<any>(this.urlApi + '/' + concertId).pipe(
    tap(() => {
      const currentConcerts = this.concertsSubject.value.filter(
        (concert) => concert.id !== concertId
      );
      this.concertsSubject.next(currentConcerts);
    })
  );
}

seleccionarConcert(concert: ConcertDTO) {
  this.concertSeleccionatSubject.next(concert);
}
}

```

El següent fragment de codi mostra el component `ConcertsComponent` definit al mòdul portal. Aquesta classe és l'encarregada de mostrar una llista actualitzada de concerts a la interfície de l'usuari. Mentre carrega la informació el component spinner es mostrarà en la pantalla, i s'ocultarà una vegada hagen carregat les dades.

```

@Component({
  selector: 'app-concerts',
  templateUrl: './concerts.component.html',
  styleUrls: ['./concerts.component.scss'],
})
export class ConcertsComponent implements OnInit {
  loading = true;
  public llista_concerts$: Observable<ConcertDTO[] | undefined> =
    new Observable<ConcertDTO[]>();

  constructor(private portalService: PortalService) {}
  ngOnInit(): void {
    this.loadConcerts();
  }

  loadConcerts(): void {
    this.portalService.getConcerts();
    this.llista_concerts$ = this.portalService.getProximsConcerts();
    this.llista_concerts$.subscribe(() => (this.loading = false));
  }
}

```

Aquest codi correspon a la pipe personalitzada 'truncaDesc', definida en el mòdul compartit, limita la longitud del text passat com a paràmetre a 50 caràcters. Si el text s'ha truncat, afegeix punts suspensius al final d'aquest.

```
@Pipe({
  name: 'truncaDesc',
})
export class TruncaDescPipe implements PipeTransform {
  transform(value: string, limit: number = 50): string {
    if (!value) {
      return '';
    }
    return value.length > limit ?
      value.substring(0, limit) + '...' :
      value;
  }
}
```

El següent codi correspon al validador de formulari personalitzat 'passwordMismatch', s'ha definit al mòdul usuari, comprova si el valor del camp 'nova_password' coincideix amb el valor del camp 'confirma_password'.

```
export function passwordValidator(): ValidatorFn {
  return (control: AbstractControl): ValidationErrors | null => {
    const confirmPassword = control.get('confirma_password');
    const password = control.get('nova_password');
    const passwordMismatchValue =
      password?.value !== confirmPassword?.value;
    if (passwordMismatchValue) {
      confirmPassword?.setErrors({ passwordMismatch: true });
    } else {
      confirmPassword?.setErrors(null);
    }
    return passwordMismatchValue ?
      { invalid: passwordMismatchValue } :
      null;
  };
}
```