



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

CAMPUS D'ALCOI

# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Politécnica Superior de Alcoy

### Desarrollo y despliegue de una aplicación web para una urbanización de vecinos

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Gregori Faus, Joan Manuel

Tutor/a: Esparza Peidro, Javier

CURSO ACADÉMICO: 2022/2023

## **Resum**

En l'actualitat, vivim en un món on la tecnologia, cada vegada més, forma part de les nostres vides: a casa, al treball, a l'escola, al supermercat, etc. Una de les tecnologies més conegudes i utilitzades és Internet, ja que ens permet comunicar-nos a distància, consultar informació, agilitzar processos o entretenir-nos, entre altres.

Esta circumstància explica com, hui en dia, tota organització, empresarial o no, i independentment de la seua grandària, disposa d'una pàgina web on es pot trobar informació sobre l'organització i on es solen donar a conèixer els servicis que s'oferixen. En el cas que ens ocupa en este treball, una urbanització d'unes 300 persones situada a les afores del municipi d'Alcoi i on les cases es troben separades unes d'altres, requeria d'una ferramenta que els ajudara amb les gestions pròpies d'una comunitat d'este tipus. La proposta a tal efecte va ser una aplicació web on es poguera publicar informació i notícies relatives a la urbanització, i que incloguera al mateix temps una intranet on el veïnat disposara d'alguns servicis privats concrets: un fòrum per comunicar-se internament, una pàgina per informar d'incidències, una altra per publicar les assemblees, i finalment una pàgina on crear enquestes participatives per recollir els vots del veïnat.

## **Paraules clau:**

- Frontend
- Backend
- WordPress
- MySQL
- Docker
- Caddy

## **Abstract**

Technology has become a crucial part of our world nowadays: be it at home, at work, at school, while shopping... The Internet is undoubtedly the most popular, due to its many uses and perks: it allows remote communication, it enables the access to huge amounts of information, it helps in improving and optimizing processes and services and it provides entertainment, among others.

Above facts explain why all organizations, running a business or not and independently of their size, use web sites to boost exposure by detailing who they are and what they do. In the use case presented in this work a scattered neighborhood community of about 300 residents in the outskirts of Alcoi needed a supporting tool to help with community management. The proposed solution was to create a web application which, on one side, would serve as an information board, and on the other, would include a set of private tools available on an intranet only to the users of the community: a forum for internal discussions, a board for reporting issues, another for publishing the official meeting minutes, and a page where to create interactive polls.

## **Keywords:**

- Frontend
- Backend
- WordPress
- MySQL
- Docker
- Caddy

# Contingut

1.	Introducció.....	5
1.1	Motivació.....	6
1.2	Estudi del mercat.....	7
1.3	Objectius del projecte .....	8
1.4	Estructura de la memòria .....	8
2.	Anàlisi i definició del problema .....	9
2.1	Actors.....	9
2.2	Requisits funcionals.....	11
2.3	Requisits no funcionals.....	14
3.	Disseny i solució del problema .....	16
3.1	Arquitectura de l'aplicació.....	17
3.2	Tecnologies utilitzades .....	18
3.3	Disseny vistes.....	23
3.4	Capa Servidor.....	27
3.4.1	Frontend (WordPress) .....	30
3.4.1.1	Desplegament.....	30
3.4.1.2	Introducció a WordPress .....	32
3.4.1.3	Temes .....	37
3.4.1.4	Barra de navegació .....	42
3.4.1.5	Vista Inici.....	44
3.4.1.6	Vista «Qui som?».....	45
3.4.1.7	Vista Notícies .....	46
3.4.1.8	Vista Restaurant .....	48
3.4.1.9	Vista Anunciants .....	50
3.4.1.10	Vista Inici Sessió.....	52
3.4.1.11	Vista Perfil d'Usuari .....	54
3.4.1.12	Vista Foro.....	56
3.4.1.13	Vista Assemblees .....	58
3.4.1.14	Vista Incidències .....	59
3.4.1.15	Vista Enquestes .....	63
3.4.1.16	Plugins.....	66

3.4.2 Backend (MySQL).....	78
3.4.2.1 Desplegament.....	78
3.4.2.2 Introducció a MySQL en WordPress.....	79
3.4.2.3 Taules.....	81
3.4.3 Reverse-Proxy (Caddy).....	83
3.4.3.1 Desplegament.....	83
3.4.3.2 Configuració.....	85
3.4.4 WP-CLI .....	86
3.4.4.1 Desplegament.....	86
3.4.4.2 Funcionament.....	87
3.5 Estructura i disseny del servidor.....	88
3.6 Backups.....	90
3.7 Seguretat .....	92
4. Resultat i tour per l'aplicació.....	99
5. Conclusió i treballs futurs .....	107
6. Bibliografia.....	108

# 1. Introducció

En l'actualitat vivim a un món totalment virtualitzat on pràcticament qualsevol organització o empresa disposa d'una pàgina web per oferir diferents serveis: vendre el seu producte, mostrar informació o agilitzar processos, entre altres. A més a més, pràcticament tota la població en Espanya té a l'abast un dispositiu amb un navegador web i una xarxa per connectar-se a Internet. Si fóra el cas contrari/Com no estem parlant d'un bé de primera necessitat, tots els ajuntaments disposen de llocs públics on trobaríem les ferramentes esmentades.

Este treball ve arrel d'una oferta de pràctiques que em va enviar el meu estimat professor Javier Esparza i em va agradar tant el projecte com les condicions en que anava a realitzar-se. En eixe moment, jo estava buscant faena i, a poder ser, amb condicions flexibles, per acabar les últimes assignatures que em quedaven. Per eixos motius, no vaig tardar en enviar-li el meu CV i, hui, amb el projecte acabat, puc dir que estic molt content del treball, com del tracte que he rebut en Baradello Gelat.

Quan estàs estudiant el Grau, un dels dubtes que et poden sorgir en relació al món de la informàtica és la manera en que es configura i es fa pública a Internet una pàgina web o una aplicació que hages programat al teu ordinador personal, ja que es treballa molt en local i tot s'executa a la teua màquina per vore si funciona correctament. Per eixa raó, no es sol tindre en compte la manera en que associarem una IP pública a un domini o com apuntarà eixa IP al nostre servidor o de quina manera es publicarà l'aplicació per a fer-la el més segura possible o com funcionen els certificats SSL o com es té en compte el número d'usuaris i funcionalitats a l'hora d'escollir les característiques d'un servidor.

## 1.1 Motivació

Com he esmentat abans, durant l'estudi del Grau, apareixen contínuament dubtes i problemes que et fan ser curiós en els àmbits que t'interessen. Tot alumne pensa en com serà el seu Treball Final de Grau. Jo sempre havia pensat que seria un treball com un altre que realitzes al llarg dels estudis, però no ha sigut el cas. Un dels motius principals que m'han portat a transformar una faena realitzada a unes pràctiques d'empresa en un TFG, ha sigut el simple fet de que, aquest, no es quedara sols com un simple arxiu PDF i uns programes executats en una màquina virtual. Que el meu treball poguera ser una aplicació que està funcionant i donant un servei a usuaris, que es pot accedir a través d'Internet i que, sobretot, es pot millorar, era una motivació extra.

Tota persona s'ha d'especialitzar en algun camp al llarg de la seua vida, però jo, durant l'estudi del Grau, sempre he pensat que un Enginyer Informàtic ha de tindre una base sòlida en tres camps: programació, xarxes i sistemes, i que millor que fer un treball que reuneix les tres. Programar, i després, configurar i desplegar una aplicació per fer-la visible en qualsevol navegador, seria una de les faenes que li ve a la ment a la gent quan es parla d'un informàtic. I és cert que quan estudies el Grau, en la meua menció, es donen molts conceptes en relació a això, però realment aprens i te n'adones dels problemes que comporta integrar diferents tecnologies quan estàs fent-ho i saps que del teu treball depenen moltes persones. Tots eixos motius han sigut més que suficients per a convertir un projecte real i existent en el treball que esteu llegint.

A més a més, este treball proporcionarà als lectors una font d'informació en diferents camps. Primerament s'aprendran molts coneixements en WordPress: els llenguatges de programació que s'utilitzen normalment com PHP, JavaScript, HTML i CSS; la seua configuració com poden ser les pàgines, els menús, els usuaris, els plugins, els temes, etc; l'organització de directoris i arxius que és molt important per saber com funciona internament; l'estructura de la base de dades i les taules que es creen per defecte... També s'aprendran molts coneixements en Docker: com desplegar contenidors i mapejar els seus directoris o ports, com automatitzar al màxim la posada en marxa de la web, com executar scripts quan el contenidor s'alça... Finalment, es voran altres tecnologies com Linux, ja que es prepararà i configurarà un servidor Ubuntu per al desplegament de l'aplicació Web o com Caddy que és un servidor web que utilitzarem com a Reverse-Proxy.

## 1.2 Estudi del mercat

En l'estudi del mercat, van a buscar-se aplicacions desplegades que siguin paregudes o que tinguin relació amb el que s'ha desenvolupat en este projecte. Per començar, la tecnologia principal del projecte és WordPress, un dels CMS més utilitzats i famosos del món. Però no sols això, s'estima que al voltant d'un 40% de les pàgines web estan fetes amb WordPress, una dada que si es pensa amb deteniment, és una bogeria. No obstant, no sols es fan pàgines web menudes amb esta tecnologia. A continuació, van a esmentar-se pàgines molt conegudes que s'han desenvolupat amb el CMS més popular del món: <https://www.beyonce.com/> , <https://www.bbc.com/> , <https://thewaltdisneycompany.com/>, <https://open.spotify.com/> , <https://www.skype.com/en/blogs/> ... I com es pot comprovar, no són pàgines web senzilles o minimalistes, i és que este CMS ens ofereix un muntó de possibilitats. Si seguim amb les tecnologies, MySQL, és un dels Sistemes Gestor de Bases de Dades relacional més utilitzats a l'hora de desenvolupar pàgines web. Per últim, per a desplegar serveis d'este tipus, solen utilitzar-se diferents distribucions de Linux i una de les més utilitzades és Ubuntu.

L'estudi del mercat enfocat al contingut de la pàgina web ens mostra les semblances que té la nostra aplicació front a altres. La pàgina web es pot dividir en dos grans parts. La primera és la part visible per qualsevol usuari d'Internet. Esta vista té moltes semblances a qualsevol altra pàgina ja que disposa d'informació sobre la urbanització, d'un mapa amb ubicació, d'un formulari de contacte, notícies i altra informació que sol tindre la majoria de pàgines web independentment de la seua finalitat. La segona part és la Intranet on sols usuaris autoritzats poden accedir. Una de les vistes de la nostra Intranet, que és normal tindre en qualsevol pàgina on accedeixen usuaris registrats, és la pàgina del Perfil on les persones poden modificar les dades del seu compte. Altra funcionalitat és el foro on es comuniquen els usuaris de la urbanització que es pot trobar en altres pàgines, entre elles la de la UPV, on els alumnes poden comunicar-se amb la seua classe o qualsevol xarxa social que permet comunicar-se als usuaris. La nostra pàgina web té un gestor d'incidències, diferent al d'altres pàgines web, però hi ha moltes altres aplicacions en el mercat per a gestionar incidències. Pel que fa a les assemblees, qualsevol organització disposa d'un apartat un publiquen informació relacionada amb coses internes on el cas més proper i paregut que es pot vore seria en la pàgina de la UPV que permet als professors carregar material i classes per a que els alumnes ho consulten. Per últim, la nostra pàgina disposa d'un apartat per a publicar enquestes i que la gent vote. En este cas, hi ha multitud de pàgines que permeten mostrar enquestes, però hui en dia, on més enquestes es poden vore, és a les xarxes socials com Twitter o Twitch. Com es pot vore, les funcionalitats que disposa la nostra pàgina web són molt comuns a qualsevol pàgina ja que són accions bàsiques que es necessiten en qualsevol comunitat o grup d'usuaris per a estar comunicats entre sí.



## 1.3 Objectius del projecte

Els objectius d'este projecte són implementar, desenvolupar i desplegar una pàgina web totalment funcional adequant-se als requisits i necessitats de l'Associació de Veïns que es descriuran a continuació.

Per a qualsevol usuari d'internet: Disposar d'un portal de notícies sobre l'associació, disposar d'una pàgina per a que els veïns anuncien els seus negocis, disposar d'una pàgina d'informació i contacte de la urbanització i disposar d'una pàgina per a publicar anuncis sobre el restaurant.

Per als usuaris de la urbanització: Disposar d'un apartat per a publicar les assemblees i informació sobre elles, disposar d'una foro per a establir fils de discussió i debat sobre diferents temes que afecten a l'associació, disposar d'una pàgina per a publicar incidències que es donen en la urbanització i que les persones indicades les puguen solucionar el més ràpid possible i, finalment, disposar d'una pàgina per a crear enquestes personalitzades sobre qualsevol tema i que els veïns puguen votar.

Tots els requisits anteriors s'han de desenvolupar pensant en que l'associació la conformen persones de totes les edats i la majoria poden tindre coneixements mínims en les noves tecnologies i, per tant, s'ha de fer un lloc web molt accessible i intuïtiu.

## 1.4 Estructura de la memòria

En este apartat va a explicar-se l'estructura que tindran els apartats de la memòria.

En el punt dos, definirem el problema esmentant els actors, els requisits funcionals i no funcionals.

En el punt tres, definirem el disseny i la solució del problema explicant l'arquitectura que tindrà el projecte, les tecnologies que van a utilitzar-se, el primer disseny dels apartats amb *mockups* senzills i també es profunditzarà en la definició de la capa servidor amb els seus contenidors WordPress, MySQL, Caddy i WP-CLI. Es seguirà amb el perquè de l'estructura i el disseny dels directoris al servidor, amb el tema de les còpies de seguretat i, finalment, amb la seguretat de l'aplicació.

El punt quatre és el resultat del projecte que consistirà amb captures de pantalla de com ha quedat.

El punt cinc consisteix en les conclusions i millores futures per al projecte.

L'últim punt, el sis, consisteix en la bibliografia.

## 2. Anàlisi i definició del problema

En este apartat van a llistar-se i detallar-se amb detall els requisits i funcionalitats de l'aplicació.

### 2.1 Actors

En primer lloc, s'han d'avaluar i analitzar els requisits per a definir els tipus d'usuaris que es necessiten i els seus permisos. A continuació, s'enumeraran els diferents rols d'usuaris que es requereixen per a la pàgina web:

- **Administradors.** Com en tota aplicació o sistema, es necessiten usuaris que tinguen permisos per a gestionar tota la pàgina web. Quan s'inicia un sistema, es crea un usuari administrador per defecte, però després se'n poden afegir més. A ser possible, estos usuaris han de ser els justos i necessaris per evitar problemes. També és important que els administradors tinguen certs coneixements en la tecnologia que administren, ja que tindran permisos per a canviar qualsevol aspecte o configuració del sistema i pot ser molt perillós en alguns casos on la persona no sap el que està fent. En este cas, els administradors tenen permís per a modificar qualsevol aspecte de la configuració del WordPress: els custom post types (notícies, incidències, anunciant i assemblees), els usuaris, els fitxers que estan pujats, les pàgines, els comentaris, els plugins i altres aspectes de la configuració que es troben al panel d'administració de WP.
- **Membres de la junta.** Com s'ha explicat abans, totes les aplicacions o sistemes necessiten usuaris administradors. En molts casos, es crea un rol d'usuari amb els permisos d'administració sols per a les funcions necessàries però sense tindre tots els permisos i evitar problemes que poden afectar al funcionament de l'aplicació. Estos usuaris són una espècie de "pseudoadministradors" i en este cas s'han creat amb permís per a gestionar (afegir, eliminar o editar) qualsevol custom post type (notícies, incidències, anunciant i assemblees), administrar els plugins (afegir categories o subforos al foro o afegir enquestes, entre altres), administrar els fitxers que es puguen a la pàgina web o administrar els comentaris principalment.

- **Manteniment.** Un dels requisits de la web és un apartat on els usuaris puguin publicar incidències que passen a la urbanització. L'usuari de manteniment podrà veure totes aquestes incidències i canviar el seu estat segons convinga. Primerament, les incidències tindran l'estat de "Proposta", el mantenidor podrà "Acceptar-la" o "Rebutjar-la". En el moment que l'accepte, la incidència canviarà l'estat a "Acceptada". Així, l'usuari que l'haja publicada podrà veure que hi ha una persona de manteniment que està solucionant-la. També pot donar-se el cas que, una vegada acceptada, el mantenidor no pugui solucionar-la i torne a "rebutjar-la". En aquest cas, la incidència tornarà a l'estat de "Proposta". Si tot va bé i es soluciona sense inconvenients, el mantenidor podrà "Tancar" la incidència i passarà a l'estat "Tancada". L'usuari de manteniment és l'únic (junt als administradors i a la junta) que pot canviar d'estat les incidències i que pot visualitzar totes les incidències, ja que als usuaris normals sols els apareixeran les incidències que han publicat ells mateixos.
  
- **Restaurant.** Altre requisit de la web és un apartat públic per a que el Restaurant de la urbanització pugui publicar els seus menús i informació. Per això, s'ha creat un perfil d'usuari especial que (junt a administradors i a la junta) podrà afegir fitxers en la pàgina del Restaurant. Aquest rol d'usuari sols tindrà aquesta particularitat.
  
- **Subscriptors** (Usuaris normals - veïns). Totes les pàgines web que disposen d'un intranet han de tindre usuaris per a utilitzar els serveis que estan oferint-se. En aquest cas, cada parcel·la de la urbanització disposa d'un usuari on el nom tindrà el format de "carrernúmero". Cal remarcar que les funcions que tenen aquests usuaris també les tenen tots els perfils d'usuaris anteriors. Aquests usuaris podran crear nous fils en el foro i participar en ells, podran enviar incidències, podran consultar el registre de les assemblees que estiga publicat i podran veure el registre i participar en les enquestes que publiqui la junta.
  
- **Qualsevol usuari d'Internet.** Per últim, cal parlar dels únics actors que no podran iniciar sessió, és a dir, que no tindran un usuari i una contrasenya, amb els permisos que tenen els actors que hem esmentat anteriorment. Aquests actors únicament podran consultar informació (les notícies que es publiquen, els anunciant, el restaurant...) i contactar amb la urbanització mitjançant un formulari de contacte.

## 2.2 Requisits funcionals

Els requisits funcionals són les necessitats reals dels actors i les funcions que tindrà l'aplicació o el sistema.

En este punt, es poden dividir els requisits funcionals en dos tipus.

Per una banda tenim les accions que podrà fer qualsevol usuari d'Internet:

- **Inici.** És la pàgina principal es mostre quan busquem el domini en qualsevol navegador. En este cas, es va decidir que la pàgina principal tindria, en el següent ordre, una secció amb una imatge coneguda en la urbanització, una secció amb les últimes notícies publicades i enllaços a elles, una secció amb una breu introducció de la pàgina "Qui som?" i un enllaç a ella, una secció amb un mapa de la urbanització i una secció amb un formulari de contacte que enviarà un missatge al correu de l'associació.
- **Qui som?.** És la pàgina que té la finalitat de donar a conèixer i explicar de què tracta la pàgina que s'està visitant. És una pàgina habitual en qualsevol lloc web. Deu ser una vista totalment dinàmica que canviarà segons els administradors vulguen i que es pot personalitzar de la manera que es necessite.
- **Notícies.** Altre requisit era tindre un apartat per a poder publicar notícies sobre l'Associació que puguen interessar tant a veïns com a gent externa. Esta serà la pàgina de Notícies, les quals es mostraran en forma de llista. Quan hi haja una gran quantitat de notícies, es requerirà facilitar la cerca, per tant es necessitarà poder filtrar tant per text com per dates. Els administradors i membres de la junta necessiten administrar les notícies de la manera més còmoda possible, per eixa raó, es disposaran opcions en cada notícia per poder editar-les i eliminar-les amb més facilitat sense haver d'entrar al panel d'administració a buscar-les.
- **Restaurant.** Altra necessitat era tindre un apartat per a que el Restaurant de la Urbanització que poguera anunciar qualsevol tipus d'informació com poden ser els menús, horaris o la carta. És un requisit molt important, ja que, hui en dia, la majoria de gent sol buscar per internet informació sobre llocs per anar a menjar i les decisions van molt relacionades amb el menjar, els preus o l'horari.

Es necessita una pàgina dinàmica, pareguda a la de “Qui som?”, que es puga personalitzar de diferents maneres, en funció de les necessitats. A més a més, es requereix que els administradors i usuaris del Restaurant tinguen permisos per a pujar fitxers, com imatges o pdf, des de la mateixa pàgina, cosa que facilitarà molt la gestió d’este apartat i permetrà als visitants descarregar o visualitzar.

- **Anunciants.** Altre requisit molt interessant que es va demanar, era tindre l’apartat Anunciants. Una pàgina per a que els veïns pogueren publicitar i donar a conèixer els seus negocis o qualsevol cosa que els interesse. Els membres de la junta o administradors seran els encarregats de publicar i gestionar el que els veïns envien.
- **Idioma.** L’últim requisit funcional, per a qualsevol usuari, era publicar la pàgina permetent als usuaris navegar tant en valencià com en castellà.

Per l’altra banda estan les accions que podran dur a terme les persones que tinguen un compte amb usuari i contrasenya registrats al sistema:

- **Inici de sessió.** Per a poder iniciar sessió, requerien d’una pàgina on pogueren introduir les seues credencials, usuari i contrasenya. A més a més, eixa pàgina deu tindre un enllaç amb opció per a enviar un correu on canviar la contrasenya en cas que l’usuari s’oblidi d’ella.
- **Perfil.** Altre requisit era tindre un apartat de Perfil d’usuari per a poder consultar informació sobre el compte o canviar dades com el correu o la contrasenya una vegada la sessió està iniciada.
- **Foro.** Una de les necessitats més important per als veïns és la de tindre una pàgina on poder obrir diferents fils de discussió per a debatre i parlar diferents temes que afecten el dia a dia de la Comunitat. En este foro, es necessita que qualsevol usuari puga publicar temes de discussió dins d’una de les categories existents. I dins del tema, es puguen publicar comentaris amb imatges per a fer el debat el més còmode possible. També, els administradors o membres de la junta requereixen configurar el foro de la manera més fàcil possible amb un accés directe des de la pàgina del foro. Es requereix que els administradors puguen configurar diferents aspectes del foro com els permisos dels usuaris, l’aspecte o les funcionalitats, entre altres.

- **Incidències.** Altra de les necessitats imprescindibles era tindre un apartat per a que els veïns publiquen incidències que ocorren a la urbanització. Les incidències poden ser qualsevol cosa: Alguna zona bruta, algun carrer fet malbé, problemes en l'enllumenat, etc. Una incidència requereix d'un títol, una descripció i arxius (imatges o pdf) i els usuaris podran afegir-ne segons creguen. També es necessita que les incidències tinguen diferents estats (Pendent, Acceptada, Tancada) per saber si està en procés de solucionar-se o està solucionada. Els administradors i els membres de manteniment necessiten veure la llista de totes les incidències i el nom de l'usuari que les ha publicades perquè són els encarregats de solucionar-les i els únics que podran canviar-li l'estat segons vaja el procés. També es necessita afegir comentaris a les incidències per aclarir o parlar de qualsevol tema relacionat.
  
- **Assemblees.** Altre requisit indispensable en associacions d'este tipus és el de publicar informació sobre les reunions, juntes o assemblees que es realitzen periòdicament per a que quede constància dels temes tractats i de les decisions preses. En este cas, es necessita una pàgina que mostre les assemblees i la data en forma de taula i que es puga accedir a cadascuna individualment per a consultar més informació i fitxers adjunts. Els administradors i membres de la junta seran els encarregats de publicar-les.
  
- **Enquestes.** L'últim requisit és una pàgina on els administradors i membres de la junta puguen publicar enquestes personalitzades. Al publicar la pregunta és menester que es puguen escriure un número de respostes variable. També és convenient publicar enquestes formatejades i amb imatges a les respostes. Altra necessitat és tindre la possibilitat d'automatitzar o fer manual la tancada de les enquestes. Esta pàgina deu mostrar una taula amb les enquestes i els diferents estats (Tancada, Pendent, Oberta). Les enquestes tancades ja han acabat el seu termini per a votar i, per tant, es poden consultar els resultats. Les enquestes pendents, estan a l'espera de que l'usuari vote. A les enquestes obertes, l'usuari ja ha votat i fins que no es tanquen, no es poden consultar els resultats.

## 2.3 Requisits no funcionals

Els requisits no funcionals són aquells que depenen en gran part del desenvolupador i que tracten temes relacionats amb el sistema que s'està implementant i les característiques que tindrà per ajustar la qualitat del software al màxim possible.

- **Còpia de seguretat.** Qualsevol sistema requereix de certes còpies de seguretat, tant del codi com de les dades. La regla 3-2-1 diu que és imprescindible realitzar tres còpies de seguretat, guardar-ne dos en llocs diferents i una còpia s'ha de mantindre fora de l'empresa. El codi de l'aplicació requereix d'una còpia de la base de dades i del codi font de l'aplicació diàries, ja que hi ha directoris on s'emmagatzemen fitxers carregats pels usuaris. I almenys es necessita una còpia de seguretat d'altres fitxers i directoris de l'aplicació, que no són modificats, però es necessiten per al funcionament.
- **Seguretat.** Hui en dia qualsevol sistema està en perill de ser atacat, així que és recomanable aplicar certes pautes de seguretat per a fer-lo el menys vulnerable possible i evitar que s'infiltrin al sistema o que ens roben dades.
- **Manteniment.** Els sistemes requereixen d'un cert manteniment. Les pàgines web sobretot necessiten actualitzacions del software per evitar "bugs" o problemes de seguretat.
- **Llançament.** Quan llancem una pàgina web, estem desplegant un conjunt de servicis. Els serveis es poden automatitzar de manera que amb una sola instrucció pugam desplegar la pàgina web. És un requisit important per evitar problemes de dependències o problemes d'haver-nos deixat alguna part del sistema per llançar.
- **Escalabilitat.** Un sistema escalable és aquell que pot créixer en vertical (afegint capes) o en horitzontal (afegint funcionalitats). Per a crear una pàgina web escalable, els serveis i funcionalitats han d'estar molt ben separades i definides. Es requereix una pàgina el més escalable possible per actualitzacions futures.
- **Portabilitat.** Un sistema portable és aquell que es pot traslladar i muntar a altres servidors de manera molt senzilla i ràpida. Si un sistema té còpies de seguretat, el seu llançament és ben senzill i, a més, està muntat en un servidor

Git de manera que es pot carregar i descarregar fàcilment, este sistema serà portable.

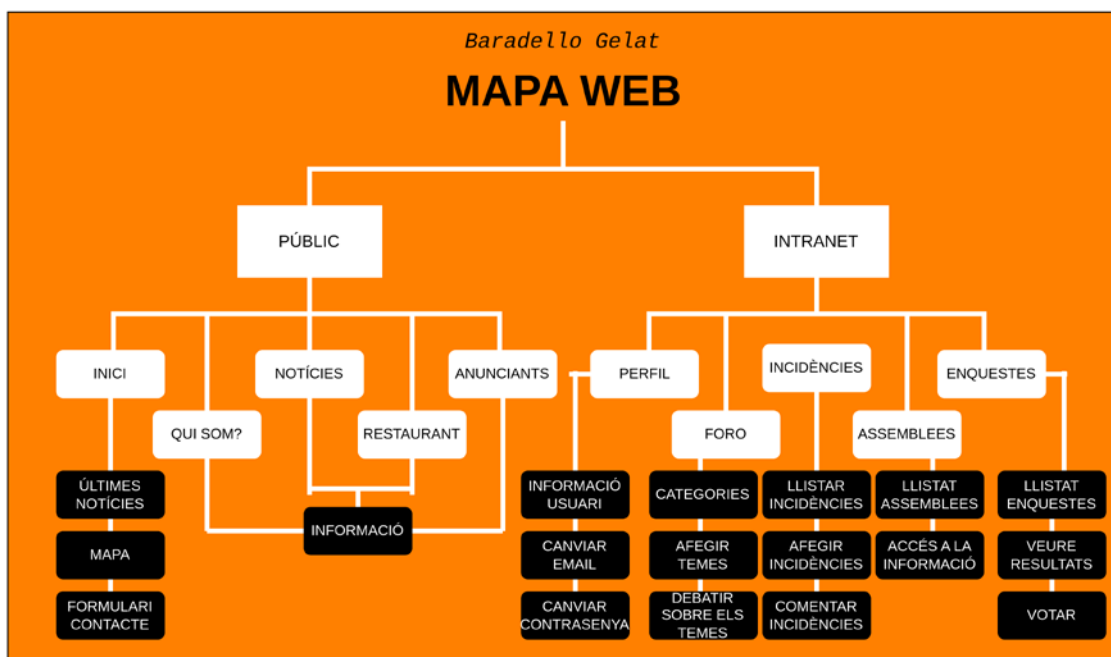
- **Compatibilitat.** Una pàgina web compatible és aquella que és accessible des de diferents sistemes operatius o navegadors. Hui en dia la compatibilitat està molt estesa i pràcticament tots els llenguatges de programació Web i tots els Sistemes Operatius estan adaptats per servir pàgines web i que siguin accessibles des de qualsevol navegador.
- **Usabilitat.** Una pàgina web usable és aquella que és comprensible i fàcil de gastar per qualsevol persona independentment dels seus coneixements en informàtica. Este requisit és molt important per a esta Urbanització ja que la majoria de gent no té coneixements en noves tecnologies i és molt important fer un sistema usable per a que tots els veïns tinguin les mateixes oportunitats per fer-lo servir.



### 3. Disseny i solució del problema

Este apartat conté tota la informació sobre com s'ha construït l'aplicació web. Primer es farà un anàlisi general de l'arquitectura, separant-la en capes. Després es descriuran les tecnologies utilitzades i el perquè. Més avant entrarem més a fons detallant l'estructura i funcionament de cada capa. Després es farà l'explicació de l'estructura i el disseny del servidor. Finalment es vorà la part dels backups i de la seguretat implementada a l'aplicació.

A continuació va a mostrar-se un mapa conceptual amb una visió molt general i esquemàtica de l'aplicació:

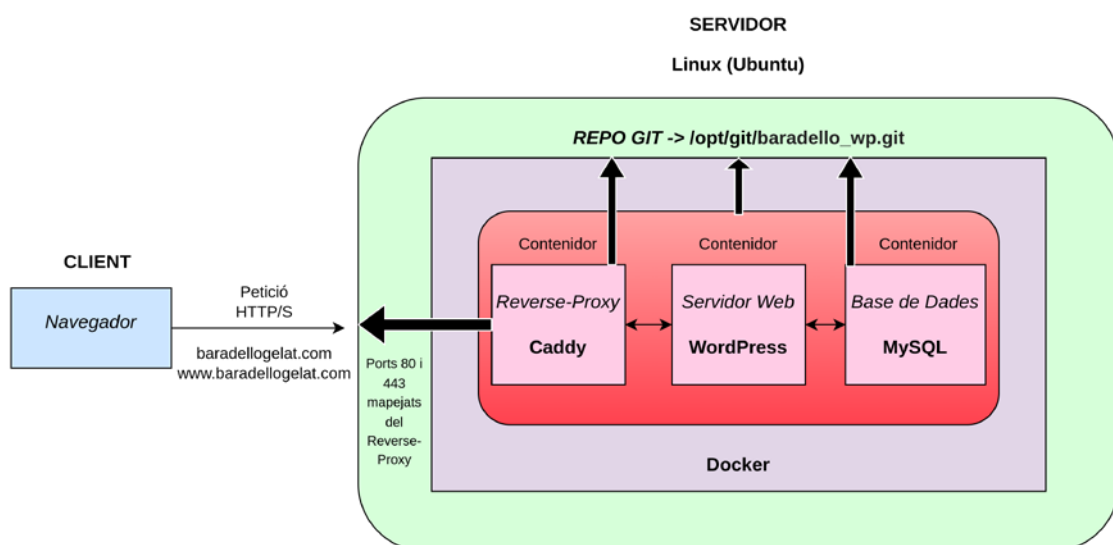


### 3.1 Arquitectura de l'aplicació

L'arquitectura web és l'estructura i el disseny tècnic dels components que conformen l'aplicació web i com es comuniquen entre ells. Hi ha moltes maneres d'enfocar i dissenyar l'estructura perquè va en funció dels objectius i requeriments i d'això dependran característiques com l'escalabilitat, la facilitat del manteniment o el rendiment, entre altres. En el nostre cas, com en moltes altres aplicacions web, s'ha escollit l'arquitectura Renderització del Costat del Servidor (Server Side Rendering). Esta tècnica tracta de que el client sol·licita al servidor una pàgina i este li l'envia ja representada a falta del codi JavaScript que l'interpreta el navegador del client.

En la part del Servidor és on tenim totes les dades i el codi de l'aplicació desplegat com a serveis per a fer-ho accessible amb peticions HTTPS. La capa del servidor està separada en tres contenidors Docker per fer-la més segura i aïllada. Els tres contenidors són, com sol ser normal, el que es coneix per "frontend" amb un contenidor WordPress, el "backend" amb un contenidor MySQL i el "reverse-proxy" amb Caddy.

El contenidor Caddy és el servei que està escoltant totes les peticions que arriben al servidor a través dels dominis `baradellogelat.com/www.baradellogelat.com` que estan apuntant a la IP Pública del Servidor. El "reverse-proxy" redirigeix estes peticions al contenidor de WordPress que és l'encarregat de mostrar tot el contingut de la pàgina web. Al mateix temps, WordPress fa servir el contenidor de la base de dades per obtenir aquella informació requerida per l'usuari i que ha de mostrar.



## 3.2 Tecnologies utilitzades

**Docker.** Docker [1] és una aplicació de codi obert que ens permet automatitzar el desplegament de contenidors software independents de la nostra màquina. En certa manera, sobre un mateix hardware, podem muntar contenidors amb el seu sistema operatiu i amb les seues aplicacions aïllades com si disposàrem de més d'un servidor.

Docker ens ofereix molts avantatges:

- La velocitat. Docker és capaç d'arrancar contenidors de manera immediata, ja que no ha d'alçar un sistema operatiu de zero sobre un hardware, com passa, per exemple, a una màquina virtual.
- Aïllament. Docker garanteix que les aplicacions i els recursos estiguen aïllats en cada contenidor. Això implica que quan eliminem un contenidor no deixa rastre ni d'arxius de configuració, ni d'arxius temporals al sistema operatiu de la nostra màquina.
- Seguretat. Les aplicacions que s'executen en contenidors docker estan completament segregades les unes de les altres. Cap contenidor pot vore els processos que s'executen en altre contenidor, és a dir, cadascun d'aquests obté el seu propi conjunt de recursos que van des del processament fins les piles de xarxa.
- Compatibilitat. Les imatges de docker s'executen igual en qualsevol servidor o computadora portàtil, sense haver de preparar un entorn específic abans.

Bàsicament, les principals característiques que han fet que s'utilitze Docker són la rapidesa i facilitat per desplegar i eliminar els contenidors, la compatibilitat que té Docker amb WordPress i MySQL i la seguretat i aïllament que ens ofereix.

**WordPress.** WordPress [2] [3] [4] és un sistema de gestió de continguts (CMS) per a crear i mantindre qualsevol tipus de web. Hi ha diferents factors pels quals es va escollir WordPress per a fer la pàgina web: és gratuït, ofereix moltes plantilles i plugins, es pot gestionar des de l'aplicació, té una migració molt senzilla, està molt ben integrat, té molta documentació, disposa d'una API REST, és molt fàcil i ràpid alçar-lo amb Docker, etc... A banda de tots els avantatges que suposa utilitzar WordPress, una d'elles és que és el CMS més popular i utilitzat arreu del món i això vol dir que tindrem molta documentació sobre la configuració, sobre problemes, sobre arquitectura... A més a més, és important remarcar que WordPress està programat en PHP per a entorns que executen MySQL i Apache encara que també es pot desplegar amb PostgreSQL, MariaDB i Nginx.

**MySQL.** MySQL és un Sistema de Gestió de Bases de Dades relacional que en este cas utilitza SQL per a obtenir les dades.

Les bases de dades són una espècies de contenidors lògics on emmagatzemem i on podem accedir a les nostres dades, existeixen els estructurats (SQL) i els no estructurats (NoSQL).

Les bases de dades SQL són relacionals, s'emmagatzemen com a text de manera estructurada en tables, on les dades registrades són les files, basant-se en el seu context, i els camps són les columnes, que són els diferents tipus d'informació. Un exemple de taula en una base de dades SQL seria el següent:

Taula Usuaris		
ID	Nom	<u>Email</u>
1	Dani	dani7290@gmail.com
2	Marta	martabg90@gmail.com

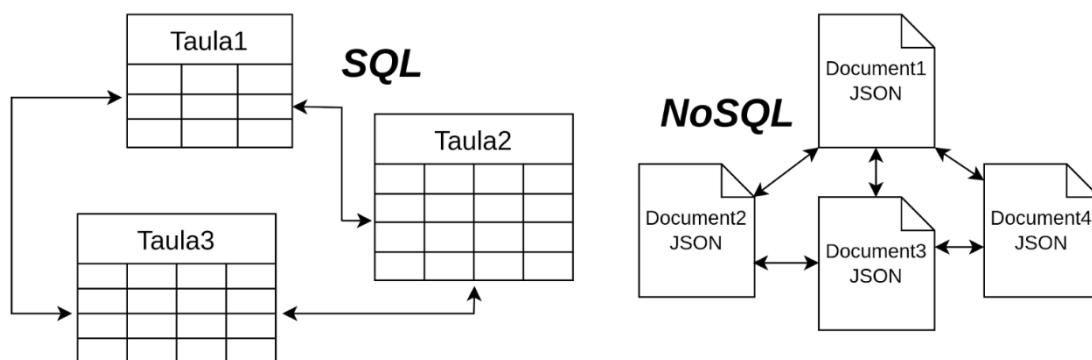
Estes taules es poden relacionar entre elles mitjançant els seus identificadors. Les dades i taules en estes BBDD es poden crear, modificar i eliminar amb llenguatge SQL (Structured Query Language). Alguns exemples de SGBD SQL són: MySQL, SQLite, Oracle, PostgreSQL, entre altres.

Per altra banda, les bases de dades NoSQL (Not Only SQL) sorgeixen davant de la necessitat per a almacenar diferents tipus d'informació no estructurada com poden ser documents, emails, àudio, vídeo, etc. Un exemple d'un document en una base de dades NoSQL podria ser el següent:

Taula Usuaris	
Clau	<u>Document</u>
1	{ "nom": "Dani", "email": "dani7290@gmail.com" }
2	{ "nom": "Marta", "email": "martabg90@gmail.com" }

Este tipus de base de dades s'utilitzen molt en el que es coneix com a Big Data perquè són molt escalables. Alguns exemples de SGDB NoSQL són: MongoDB, Redis, Neo4j, entre altres.

Les principals diferències entre les SQL i les NoSQL és que les primeres almacenen les dades de manera estructurada i les segones ho fan en el format original. També és important destacar que les NoSQL són molt més escalables que les SQL, per això s'utilitzen en el Big Data on s'emmagatzemen molta informació, i per això, les SQL estan centralitzades i les NoSQL estan distribuïdes. Així també, les SQL tenen integritat en les dades, al contrari que les NoSQL, que al buscar rapidesa, no fan èmfasi en eixe aspecte. Les dades estructurades són més fàcils d'analitzar que les no estructurades però estes últimes són més flexibles i permeten canvis més senzills en la seua arquitectura que les estructurades.



MySQL va integrat directament en WordPress i sol utilitzar-se, junt a MariaDB, com al Sistema Gestor de Bases de Dades, per defecte. També estaria formant part d'un sistema LAMP, que és un acrònim de Linux, Apache, MySQL i PHP/Python. LAMP s'utilitza per a definir l'infraestructura d'un servidor web que, en este cas, definiria també molt bé la nostra aplicació: Linux com a sistema operatiu del servidor, Apache com a servidor web (Nosaltres utilitzarem Caddy com a reverse-proxy), MySQL com a gestor de la base de dades i PHP com a llenguatge de programació principal. Moltes aplicacions web famoses utilitzen MySQL com Twitter, LinkedIn, Facebook, etc.

**PHP.** PHP és un llenguatge de programació de codi obert molt utilitzat en l'entorn web ja que permet ser incrustat amb codi HTML, és a dir, que un fitxer PHP pot contindre codi HTML com pràcticament tots els arxius de WordPress. Altra característica molt important és que PHP s'executa al servidor al contrari que JavaScript que s'executa al navegador del client i això implica que el client no podrà veure mai el codi PHP, sols podrà veure el resultat de l'script que s'ha executat. Per exemple, si l'script era un "echo 'Hola' ", l'usuari sols vorà per pantalla eixe Hola i no vorà tot el codi. De totes maneres, la raó principal per la qual s'està utilitzant PHP és perquè WordPress està basat en PHP, així que per entrar a fons d'este CMS és inevitable conèixer este llenguatge.

**JavaScript.** JavaScript és un llenguatge de programació lleuger que, al contrari que PHP, principalment s'utilitza en la part del client permetent pàgines web dinàmiques i millores en les vistes d'usuari, encara que també es pot implementar en la part del servidor. Es defineix com un llenguatge interpretat, orientat a objectes, basat en prototips, amb tipat dèbil i dinàmic. Es venia utilitzant junt a HTML en la part del client, ja que tots els navegadors moderns poden interpretar-lo, però amb l'aparició de tecnologies com AJAX o Node.js, també s'utilitza per a enviar i rebre informació del servidor. Hui en dia, és difícil imaginar una pàgina web que no tinga codi JavaScript, ja que sol anar molt lligat a HTML i CSS.

**HTML.** Com el seu nom indica és un Llenguatge de Marques d'Hipertext. Este llenguatge utilitza marques per a fer referència als diferents tipus de continguts que existeixen. Les etiquetes es defineixen dins de "<" i ">". Està considerat el llenguatge web més important i conegut en el món ja que tots els navegadors han adoptat este estàndard. HTML sol anar junt a CSS que li dona estil i junt a JavaScript que li dona funcionalitat.

**CSS (Cascading Style Sheets).** Fulles d'estil en cascada és un llenguatge que s'utilitza per a donar-li estil a fitxers HTML o XML. Separa i li dona forma al contingut de les interfícies en els documents XML. Per eixa raó, és molt utilitzat en entorns Web i seria molt estrany vore un document HTML que no tinga el seu fitxer CSS assignant-li un disseny.

**Bootstrap.** [5] És una biblioteca o conjunt de ferramentes multiplataforma (framework) que conté moltes plantilles i elements amb diferents estils basats en HTML i CSS que sols s'ocupa de la part del front-end. Bootstrap conta amb molts elements als quals se li poden donar diferents estils com contenidors, tipografies, imatges, taules, formularis, botons, barres de navegació, entre altres. Encara que ens ofereix un gran ventall de dissenys, un dels principals avantatges és la gran adaptabilitat (resposive) dels seus components que permet fer una aplicació web accessible per a qualsevol dispositiu. Bootstrap és framework molt útil en qualsevol desenvolupament web ja que és molt fàcil afegir la llibreria i molt fàcil gestionar-la.

**Caddy.** [6] És un servidor web multiplataforma i de codi obert escrit en Go. És relativament nou, però és molt conegut i utilitzat degut a la seua configuració tan senzilla i al seu ús com a servidor HTTP, a l'automatització del TLS i les aplicacions PKI. A més a més, també sol utilitzar-se com a reverse-proxy per a reenviar dades i connexions a altres servidors. La seua configuració es realitza mitjançant un fitxer JSON anomenat Caddyfile [7] que és molt més fàcil de comprendre que altres fitxers de configuració d'altres servidors web com poden ser Nginx o Apache.

**Git.** [8] És un software per a controlar versions de codi pensant en el manteniment i la eficiència de les aplicacions. Git s'utilitzarà per emmagatzemar tot el codi de la nostra aplicació i tindre la versió en producció separada de la versió de testeig. Es crearà un repositori git en el servidor de l'aplicació per carregar el codi del projecte.

### 3.3 Disseny vistes

Este apartat conté els primers *mockups* que s'han fet sobre la pàgina web per establir un consens i establir unes bases de com es podria veure la part del client.

#### Pàgina inici:

Baradello Gelat	<u>Inici</u>	Qui som?	Notícies	Restaurant	Anunciants	Iniciar sessió
Imatge						
Últimes notícies						
Qui som? Saber més						
Mapa						
Contacte Nom: Email: Text:						

#### Pàgina notícies:

Baradello Gelat	Inici	Qui som?	<u>Notícies</u>	Restaurant	Anunciants	Iniciar sessió
Notícies						
<input type="text"/>						Buscar
<input type="checkbox"/>	<b>Notícia 1</b>	resum	saber més...			
<input type="checkbox"/>	<b>Notícia 2</b>	resum	saber més...			



## Anunciants:

Baradello Gelat	Inici	Qui som?	Notícies	Restaurant	<u>Anunciants</u>	Iniciar sessió
Anunciants						
<input type="text"/>						Buscar
<input type="text"/>		<input type="text"/>		<input type="text"/>		
Anunciant1		Anunciant2		Anunciant3		
Resum		Resum		Resum		

## Pàgina inici sessió:

Baradello Gelat	Inici	Qui som?	Notícies	Restaurant	Anunciants	<u>Iniciar sessió</u>
<input type="text" value="Nom d'usuari"/>						
<input type="text" value="Contrasenya"/>						
<a href="#">He oblidat la contrasenya</a>						
<input type="button" value="Entrar"/>						

## Pàgina foro:

Baradello Gelat	Inici	Qui som?	Notícies	Restaurant	Anunciants	usuari	Tancar sessió
						Perfil	
						Foro	
						Assemblees	
						Incidències	
						Enquestes	
<input type="text" value="Foro &gt;"/>							
<b>CATEGORIA PRINCIPAL</b>							
<input type="text" value="CATEGORIA 1"/>							
<input type="text" value="CATEGORIA 2"/>							
<input type="text" value="CATEGORIA 3"/>							

## Pàgina assemblees:

Baradello Gelat	Inici	Qui som?	Notícies	Restaurant	Anunciants	usuari	Tancar sessió
						Perfil	
						Foro	
						Assemblees	
						Incidències	
						Enquestes	
<input type="text"/>					Buscar		
<b>Assemblees</b>			<b>Data</b>	<b>Acció</b>			
Assemblea 1			03/07/2022	Vore			
Assemblea 2			05/02/2022	Vore			
Assemblea 3			12/01/2021	Vore			

## Pàgina incidències:

Baradello Gelat    Inici   Qui som?   Notícies   Restaurant   Anunciants   **usuari**   Tancar sessió

Perfil  
Foro  
Assemblees  
Incidències  
Enquestes

Buscar    Nova incidència

Incidència	Autor	Data	Estat	Acció
Incidència1	usuari1	23/02/2023	Proposta	Vore
Incidència2	usuari2	10/11/2022	Acceptada	Vore
Incidència3	usuari3	21/10/2022	Completada	Vore

## Pàgina enquestes:

Baradello Gelat    Inici   Qui som?   Notícies   Restaurant   Anunciants   **usuari**   Tancar sessió

Perfil  
Foro  
Assemblees  
Incidències  
Enquestes

ID	Enquesta	Estat	Acció
1	Enquesta1	Pendent/Activa/Tancada	Vore/Votar
2	Enquesta2	Pendent/Activa/Tancada	Vore/Votar
3	Enquesta3	Pendent/Activa/Tancada	Vore/Votar

Baradello Gelat    Inici   Qui som?   Notícies   Restaurant   Anunciants   **usuari**   Tancar sessió

Perfil  
Foro  
Assemblees  
Incidències  
Enquestes

Pregunta

Opció 1    Imatge

Opció 1    Imatge

Opció 1    Imatge

Votar

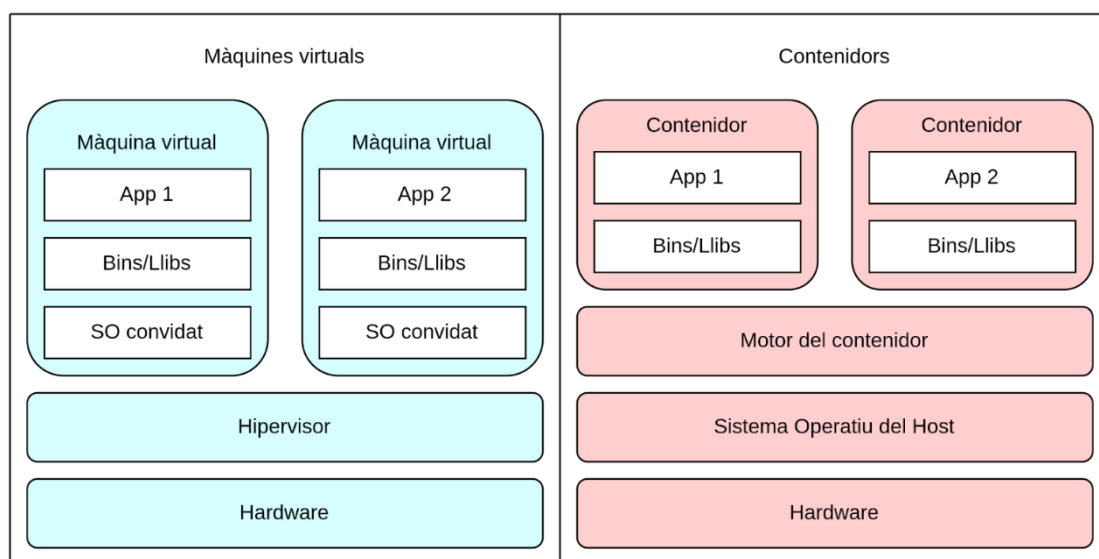
### 3.4 Capa Servidor

La capa del Servidor és la que s'encarrega de tindre el codi de l'aplicació executant-se com a servei visible i accessible a Internet, normalment, a través d'un domini que redirecciona a una IP Pública i port. També ha de tindre les dades de l'aplicació, però en este cas ben aïllades i protegides de qualsevol usuari no autoritzat per evitar problemes. Tot açò ha d'estar desplegat i ben montat a la capa servidor per a fer que funcione correctament.

Esta aplicació s'ha construït sobre un servidor Ubuntu (VPS). Per a fer més segura l'aplicació i el servidor, cadascun dels serveis que s'han fet servir estan montats en contenidors docker.

Els contenidors són software lleuger que conté totes les dependències necessàries per a virtualitzar la aplicació software que contenen. A diferència de les màquines virtuals que són software pesat que proporcionen una emulació completa del hardware a baix nivell com la CPU, el disc, el dispositius de xarxa, etc.

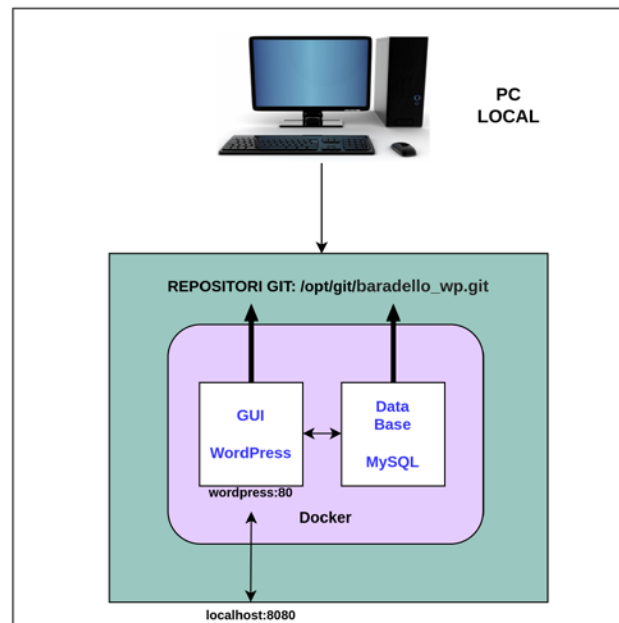
La principal diferència entre una màquina virtual i un contenidor és que la primera virtualitza fins i tot les capes hardware i, pel contrari, els contenidors sols virtualitzen les capes software per damunt del sistema operatiu.



Docker té la ferramenta Compose amb la qual podem llançar múltiples contenidors mitjançant un fitxer amb l'extensió "yml" o "yaml" on es defineix tota la configuració que volem. YAML és un format de codificació de dades llegible per persones. Va ser creat per definir llistes, mapejos o valors senzills amb cometes. Per a més informació sobre este tipus de fitxers, visitar [9]. Docker Compose [10] és una ferramenta molt útil que ens permet amb una sola instrucció alçar tot el nostre sistema i, per eixa raó, s'ha escollit per al nostre projecte.

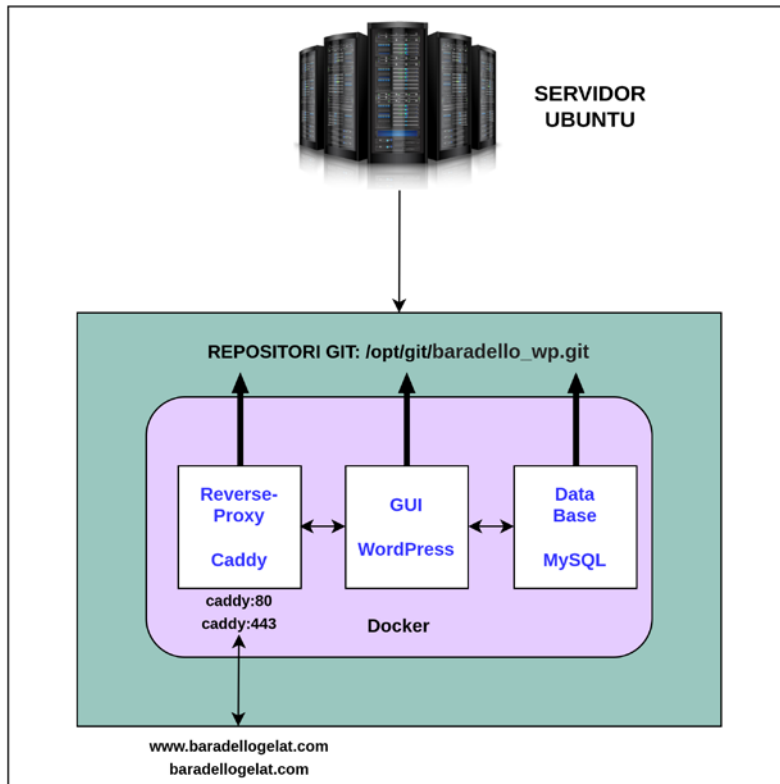
Com ja s'ha mencionat abans, esta capa es compon de tres subcapes, com, normalment, qualsevol aplicació web. No obstant, quan es desenvolupa una pàgina web, es fa en un entorn de prova controlat que, en este cas, no necessitava d'una de les capes que afegeix seguretat a l'aplicació, com és la del Reverse-Proxy.

A continuació, es mostra un esquema de l'entorn de prova mentre s'estava construint l'aplicació en un ordinador personal:



Quan l'aplicació ja estava completada en l'entorn de prova, era moment de pensar en la migració i el desplegament al Servidor Privat Virtual. La migració va ser senzilla amb el Software Git ja que es va crear un repositori per a carregar i descarregar el projecte allà on necessitem. El desplegament va comportar més problemes i reflexions. Finalment, es va decidir que la forma més segura era amagar els contenidors de WordPress i MySQL per evitar exposicions i llançar un contenidor Caddy que funcionara de Proxy Invers per a que es fera càrrec de les peticions i balancejara les càrregues augmentant així el rendiment de l'aplicació.

D'esta manera va quedar el servidor amb el contenidor Caddy:



### 3.4.1 Frontend (WordPress)

El frontend, en una aplicació web, fa referència a tot aquell codi HTML, CSS i JavaScript que crea les vistes amb les quals interactua l'usuari. En este cas, el frontend està desenvolupat amb WordPress que també conté codi PHP per a interactuar amb el backend.

En el punt 3.2 ja s'ha explicat què és exactament WordPress i els avantatges d'utilitzar-lo. Així que en este apartat entrarem més a fons: s'explicarà la configuració del fitxer YML per a desplegar el contenidor WordPress en Docker, s'explicarà quins són els directoris i arxius de WordPress, què són els temes i com s'ha desenvolupat el nostre propi tema, què són els plugins i com s'ha desenvolupat el nostre propi plugin.

#### 3.4.1.1 Desplegament

En l'arxiu docker-compose.yml es defineixen els contenidors com a serveis. En este cas tenim el servei wordpress on es defineixen els següents paràmetres:

- **container\_name.** Un nom per al contenidor fàcil d'identificar per si necessitem consultar qualsevol cosa.
- **image.** Ací es defineix la imatge i versió que el contenidor descarregarà dels repositoris de Docker Hub. En este cas s'està utilitzant l'última versió de WordPress.
- **ports.** Ací es defineixen els ports que seran mapejats per a poder vore, des del nostre equip, els serveis que estan dins del contenidor. S'està mapejant el port 80 del contenidor (on està executant-se el servei de WordPress) al port 8443 del nostre equip (s'ha escollit un port no reconegut per evitar problemes).
- **volumes.** Este paràmetre és important, ja que, quan llancem el contenidor, estem mapejant els directoris de WordPress del contenidor als directoris del nostre equip on tenim tot el codi que s'ha programat anteriorment. Açò té diferents avantatges com per exemple que al mapejar els volums correctament, l'aplicació ja està agafant el codi i no és necessari copiar-lo o utilitzar Dockerfiles o també que al fer qualsevol canvi en estos directoris s'estarà aplicant directament als directoris del contenidor, és a dir, a l'aplicació WordPress. Per mapejar volums correctament, s'ha de saber on es crea el directori arrel d'una aplicació. En este cas el projecte de WordPress està en el directori /var/html/www del contenidor. La raó per la qual s'ha separat el

directori wp-content de la resta del projecte, s'explicarà més avant, en el punt on es parla de l'estructura de carpetes.

- **environment.** Es defineixen quatre variables d'entorn molt importants com són el host, l'usuari, la contrasenya i el nom de la base de dades. El host no és important xifrar-lo però l'usuari, la contrasenya i el nom, per seguretat, s'han definit com a variables d'entorn de Linux en un fitxer .env per a que no siguin visibles a l'arxiu YML.
- **depends\_on.** Este paràmetre indica que el contenidor depen del servei db per a alçar-se. Així que fins que no s'alce db, wordpress no s'alçarà.
- **restart.** Indica que sempre que caiga es tornarà a alçar el contenidor.

```
wordpress:
  container_name: baradello_wp
  image: wordpress:latest
  ports:
    - 8443:80
  volumes:
    - ./production/www:/var/www/html
    - ./wp-content:/var/www/html/wp-content
  environment:
    WORDPRESS_DB_HOST: db
    WORDPRESS_DB_USER: ${DB_USER}
    WORDPRESS_DB_PASSWORD: ${DB_PASSWORD}
    WORDPRESS_DB_NAME: ${DB_NAME}
  depends_on:
    - db
  restart: always
```





I el directori més important pels desenvolupadors, wp-content/:

```
joanma@vmi1026407:/baradello/baradello_wp$ ll wp-content/
total 44
drwxrwxr-x 8 www-data www-data 4096 Jul 3 08:18 ./
drwxrwxr-x 7 joanma joanma 4096 Mar 9 15:18 ../
drwxrwxr-x 4 www-data www-data 4096 Dec 13 2022 .history/
-rwxr-xr-x 1 root root 28 Feb 21 00:31 index.php*
drwxrwxr-x 4 www-data www-data 12288 Feb 21 00:31 languages/
drwxrwxr-x 11 www-data www-data 4096 Jul 3 08:18 plugins/
drwxrwxr-x 8 www-data www-data 4096 Jul 3 08:18 themes/
drwxrwxr-x 2 www-data www-data 4096 May 20 07:39 upgrade/
drwxrwxr-x 6 www-data www-data 4096 Jan 1 2023 uploads/
```

El directori dels plugins:

```
joanma@vmi1026407:/baradello/baradello_wp$ ll wp-content/plugins/
total 52
drwxrwxr-x 11 www-data www-data 4096 Jul 3 08:18 ./
drwxrwxr-x 8 www-data www-data 4096 Jul 3 08:18 ../
drwxrwxr-x 4 www-data www-data 4096 Feb 21 00:31 akismet/
drwxrwxr-x 2 www-data www-data 4096 Mar 6 14:29 baradello/
drwxrwxr-x 3 www-data www-data 4096 Feb 21 00:31 classic-editor/
drwxrwxr-x 5 www-data www-data 4096 Feb 21 00:31 gmail-smtp/
drwxrwxr-x 5 www-data www-data 4096 Feb 21 00:31 gtranslate/
-rwxr-xr-x 1 root root 2578 Feb 21 00:31 hello.php*
-rwxr-xr-x 1 root root 28 Feb 21 00:31 index.php*
drwxrwxr-x 5 www-data www-data 4096 Jan 19 23:24 jwt-authentication-for-wp-rest-api/
drwxrwxr-x 11 www-data www-data 4096 Dec 13 2022 wpforo/
drwxrwxr-x 7 www-data www-data 4096 Feb 21 00:31 wp-mail-smtp/
drwxrwxr-x 4 www-data www-data 4096 Feb 23 00:13 wp-polls/
```

El directori del tema actiu BootScore-Child:

```
joanma@vmi1026407:/baradello/baradello_wp$ ll wp-content/themes/bootscore-child-main/
archive-advertiser.php  comments.php  front-page.php  3s  page-perfil.php  README.md  scss  search.php  single-advertiser.php  style.css
archive-assembly.php  css  Functions.php  LICENSE  page.php  readme.txt  searchform-assembly.php  sidebar-assembly.php  single-assembly.php  template-parts
archive-issue.php  footer.php  header.php  login-page.php  page-restaurant.php  recover_password_page.php  searchform-issue.php  sidebar-issue.php  single-issue.php
archive.php  frontpage  img  old-page-issue.php  profile-page.php  screenshot.png  searchform.php  sidebar.php  single.php
```

A continuació van a explicar-se els fitxers i carpetes amb més importància d'un projecte WordPress.

En primer lloc, estan els fitxers per defecte del directori arrel en un projecte WordPress:

**.htaccess.** És un fitxer de configuració del servidor Web Apache. En el cas que el WordPress el tinguem allotjat en un altre servidor web com Nginx, este fitxer no apareixerà. Este arxiu indica al servidor web com actuar mitjançant regles, encara que, per defecte, controla els enllaços permanents de WordPress. A més a més, es poden configurar redireccions automàtiques, restriccions de seguretat per a certs arxius o IP o altres temes relacionats en la configuració del servidor web.

**wp-activate.php.** És un fitxer que confirma que la clau d'activació que s'envia en un correu electrònic, després de que un usuari es registre, coincideix amb la clau d'este usuari i mostra la confirmació.

**wp-blog-header.php.** És un fitxer que carrega l'entorn i la plantilla de WordPress.

**wp-comments-post.php.** És un fitxer per a controlar la publicació de comentaris en WordPress i evitar els comentaris duplicats.

**wp-config.php.** Un fitxer imprescindible, ja que conté molts aspectes de la configuració bàsica de WordPress. Per exemple, este arxiu conté el nom, l'usuari, la contrasenya i el host, el charset i el collate de la base de dades. També conté altres variables com: huit claus de seguretat que es generen al fer la instal·lació, el prefix de les taules de la base de dades, la ubicació o estructura de les carpetes si la volem modificar o si volem posar el mode debug i logs, entre altres.

**wp-cron.php.** És un arxiu de cron que s'activa cada vegada que la pàgina web reb una visita o quan es crida a través d'un servidor cron daemon.

**wp-load.php.** És un fitxer de Bootstrap per a configurar la variable ABSPATH i carregar l'arxiu wp-config.php que després carregarà l'arxiu wp-settings.php

**wp-login.php.** És la pàgina de l'usuari en WordPress ja que controla aspectes com l'autenticació, el registre, la recuperació de contrasenya i altres coses relacionades amb l'usuari.

**wp-settings.php.** És un fitxer per a configurar i corregir variables comuns i incloure les biblioteques de procediments i classes de WordPress

**wp-signup.php.** És un fitxer per a configurar l'entorn de WordPress

En el directori arrel de WordPress trobem tres subdirectoris molt importants:

**wp-admin.** Este directori és el que conté tots els arxius i carpetes que formen el panel d'administració de WordPress. És un dels directoris més importants ja que el panel wp-admin facilita molt la configuració de la nostra pàgina web i ajuda a que qualsevol persona sense coneixements de programació pugui gestionar una aplicació d'aquestes característiques.

**wp-includes.** Este directori és molt important en WordPress també, ja que conté fitxers per a fer funcionar tant el backend com el frontend de l'aplicació. Altres carpetes i arxius requereixen de les funcions d'aquest directori. A més a més, conté gran part del codi font de WP i també conté la funcionalitat que necessita l'API REST de WordPress per a funcionar.

**wp-content.** És la carpeta més important des del punt de vista del desenvolupador, ja que en este directori és on es faran la majoria dels canvis des del punt de vista del programador. Este directori conté els plugins, els temes i els fitxers que els usuaris carreguen a la pàgina. En la carpeta plugins es mostren tots els plugins que estan instal·lats a l'aplicació. El mateix amb els temes, en la carpeta themes es veuen tots els temes instal·lats. En la carpeta uploads, estan els directoris dividits en anys i dins dividits en mesos així que és molt fàcil d'administrar este aspecte també.

Ara que ja s'han descrit les parts més importants d'un projecte WordPress, també és important descriure el funcionament intern i la jerarquia de les plantilles [11]. WordPress utilitza la cadena de consulta, que és informació continguda en l'enllaç a cada part de la pàgina web, per decidir quines plantilles s'han de mostrar.

Per determinar les plantilles, WordPress:

1. Fa coincidir una cadena de consulta amb un tipus de consulta per decidir quina pàgina es sol·licita (pàgina de cerca, de categoria, etc.)
2. Selecciona la plantilla en l'ordre determinat de jerarquia.
3. Busca arxius de plantilla amb noms específics en el directori del tema actual i utilitza el primer en coincidir.

Quan s'utilitza un tema secundari, qualsevol arxiu que s'agregue al tema secundari, anul·larà el mateix fitxer del tema principal. Si els dos temes contenen la plantilla archive.php, s'utilitzarà la del tema secundari.

Si el nostre blog està a la pàgina <https://baradellogelat.com/blog> i volem visitar una categoria com <https://baradellogelat.com/blog/category/lilibres>, WordPress fa el següent procediment:

1. Busca un arxiu amb el directori del tema actual: category-llibres.php.
2. Si category-llibres.php no està i l'ID de la categoria és el 2, es busca un fitxer cridat caregory-2.php.
3. Si no existeix, es buscarà la plantilla genèrica archive.php.
4. Si la plantilla genèrica no existeix, s'utilitzarà l'arxiu de plantilla del tema principal, index.php.

La pàgina d'inici per mostrar les últimes publicacions del blog és home.php. Si no existeix s'utilitza index.php.

La plantilla de la pàgina principal és front-page.php que té prioritat sobre home.php. Si front-page.php no existeix, es buscaran home.php o page.php depenent de la configuració en Configuració -> Lectura. Si no existeix cap arxiu, s'utilitzarà index.php.

En quant a la plantilla de publicació única, es segueix el següent ordre:

1. single-{post-type}-{slug}.php.
2. single-{post-type}.php.
3. single.php
4. singular.php
5. index.php

L'arxiu de plantilla utilitzat per representar una pàgina estàtica segueix el següent ordre:

1. Arxiu de plantilla personalitzat
2. page-{slug}.php
3. page-{id}.php
4. page.php
5. singular.php
6. index.php

Les pàgines d'etiquetes utilitzen el següent ordre:

1. tag-{slug}.php
2. tag-{id}.php
3. tag.php
4. archive.php
5. index.php

Este sistema de jerarquies permet canviar-la en punts específics. El filtrat, que està en la funció `get_query_template()`, utilitza este nom per a filtrar: “`{$type}_template`” on `$type` és el tipus de plantilla. Tots els filtrats disponibles a continuació: `embed`, `embed_template`, `404_template`, `search_template`, `frontpage_template`,

home\_template, privacypolicy\_template, taxonomy\_template, attachment\_template, single\_template, page\_template, singular\_template, category\_template, tag\_template, author\_template, date\_template, archive\_template, index\_template.

### 3.4.1.3 Temes

El responsable de l'aparença i del comportament d'una pàgina web feta en WordPress és el tema actiu. La renderització de les pàgines que li apareixen al client es fa mitjançant un procés d'uns fitxers que estan al servidor, que són els arxius de plantilles, els quals componen el tema. Són arxius PHP que contenen codi HTML, PHP o etiquetes de plantilla. Les etiquetes de plantilla són una mescla de codi HTML i PHP que es criden mitjançant funcions de WordPress [12] per obtenir algun tipus d'informació (exemple: `get_header()` o `get_footer()`).

Els arxius de plantilla són un conjunt de fitxers que componen un tema en WordPress i es desglossaran més avall.

Els temes, en WordPress, són una part imprescindible d'este tipus de pàgines web. Un tema és un conjunt de fitxers que representa el disseny i les funcionalitats de la nostra interfície web. El arxius de cada tema es troben al directori `wp-content/themes` i solen ser els arxius que s'han definit en la capa Vista. Hi ha temes gratis i temes que costen diners. Els temes gratis solen ser molt minimalistes i contenen pocs plugins i funcionalitats.

La millor opció per a un desenvolupador de WordPress, és crear el seu propi tema en WordPress i d'esta manera programar al 100% l'aplicació des de zero. No obstant, també hi ha altres opcions. Per exemple, es pot desenvolupar un tema a partir d'un altre tema amb certes llibreries o configuracions per a començar amb algunes funcionalitats i tindre una base. Este tipus de temes, en WordPress, s'anomenen temes fills.

Un tema fill, està basat i depèn d'un tema pare però té l'avantatge que tots els canvis es fan en el fill i no es modifica l'original. El temes fills funcionen de la següent manera: Primer es busca si l'arxiu està en el tema fill i si no està, es busca al tema pare.

Per a este projecte s'ha utilitzat `bootScore` que és un "starter-theme" de codi lliure per a WordPress basat en `Bootstrap 5`. Este software ja està preparat amb el tema principal i el tema fill per a desenvolupar directament sobre este últim. S'ha escollit un tema basat en `Bootstrap` per la quantitat de components que té esta llibreria que faran una interfície molt més amigable i interactiva. També és important destacar una característica molt important que s'ha de buscar en les pàgines web i és que siga

adaptable (responsive) a qualsevol dispositiu amb navegador web i els components de Bootstrap ho són.

Per tant, escollir un tema basat en Bootstrap per a desenvolupar sobre ell mateix, és molt bona opció, ja que estem integrant components, que són adaptables a qualsevol dispositiu, a una web i, a més a més, tenim una llibreria amb molts elements variats, dotats d'un cert disseny, que ens faran la nostra aplicació molt més amigable.

Tots els temes tenen, per defecte, certes plantilles que donen una vista inicial sense necessitat d'implementar cap cosa. Eixes plantilles són les següents:

- **404.php.** És la plantilla per a quan es produeix un error 404. L'error 404 és un fallo del client que indica que el navegador no pot trobar allò que està buscant-se al servidor.
- **archive.php.** És la plantilla que llista les entrades (posts) mitjançant una consulta que pot ser modificada al nostre gust ja siga per categories, per tipus o per altres característiques. Es pot personalitzar per a cada tipus de post amb `archive-{tipus_de_post}.php`
- **comments.php.** És una plantilla s'ha d'incloure dins de pàgines o altres plantilles per a visualitzar-la amb la funció `comments_template()` i mostra, en el cas que en tinga, els comentaris d'una entrada i, a més, un caixetí per a afegir-ne si és el cas.
- **header.php.** És la capçalera de les pàgines. Normalment inclou la barra de navegació personalitzada i també pot incloure alguna imatge. És una plantilla que també s'ha d'invocar des d'altres plantilles amb la funció `get_header()`.
- **footer.php.** És el peu de pàgina. Normalment inclou una espècie de menú per a navegar però sol ser molt més senzill que la capçalera. També és una plantilla que s'ha d'invocar des d'altres plantilles amb la funció `get_footer()`.
- **index.php.** És la plantilla principal d'un tema que, en principi, no es gasta a sovint però es mostrarà quan es faça una cerca i no es trobe cap plantilla dins de la jerarquia de plantilles.
- **page.php.** És la plantilla per defecte de les pàgines que creem en el panel de configuració de WordPress. Es poden crear més plantilles de pàgines amb page-

{nom\_plantilla}.php i utilitzar-les com vulgam. D'esta manera es poden personalitzar les pàgines tant com necessitem. És important saber que per a que WordPress el reconega com a plantilla s'ha d'afegir un comentari al principi del fitxer com el següent:

```
<?php
/**
 * Template Name: Nom de la plantilla
 */
```

- **single.php.** És la plantilla per defecte per a quan volem visualitzar una entrada o post. Es pot personalitzar per a cada tipus de post amb la forma single-{tipus\_de\_post}.php
- **category.php.** Esta plantilla s'utilitza per a mostrar llistes de les entrades per categoria. Es poden personalitzar amb category-{etiqueta}.php o category-{id}.php
- **author.php.** Esta plantilla mostra les entrades per autors. Es poden crear plantilles per a autors concrets segons el nom o l'id amb author-{nom}.php o author-{id}.php
- **tag.php.** Esta plantilla mostra les entrades per etiquetes. Es poden crear plantilles concrets segons el nom d'etiqueta o l'id amb tag-{slug}.php o tag-{id}.php
- **taxonomy.php.** Esta plantilla mostra les entrades per taxonomia. Es poden crear plantilles concrets segons el nom de la taxonomia amb taxonomy-{taxonomia}.php
- **date.php.** Esta plantilla mostra les entrades segons la seua data.
- **search.php.** Esta plantilla mostra els resultats de la búsqueda de les entrades i les pàgines



- **searchform.php.** Esta plantilla serveix per al formulari de búsqueda d'entrades i pàgines. S'utilitza afegint-la a altres plantilles.
- **sidebar.php.** Esta és la plantilla de la barra de navegació lateral. S'utilitza afegint-la a altres plantilles.
- **functions.php.** Este fitxer és un dels més importants del tema ja que és on es creen funcions que després es poden utilitzar en les plantilles i configuracions del tema.

Altre element que sol utilitzar-se bastant en WordPress són els widgets que són trossos de codi que es poden afegir en algunes parts de la pàgina web per a sumar característiques o elements com poden ser formularis, galeries, entre altres.

WordPress té l'arxiu `class-wp-widget.php` on es defineixen els widgets que sol estar en el directori `wp-includes`. Per a crear un nou widget personalitzat s'ha de crear una classe que estenga a la classe `WP_Widget` que és natural de WordPress. Existeixen 18 funcions dins d'esta classe però en són 4 les bàsiques que s'explicaran a continuació:

- **\_\_construct():** En esta funció és on definim el títol i la descripció del widget.
- **widget():** Esta funció és on es defineix el que va a mostrar-se per pantalla quan afegim el widget.
- **form():** És la funció on es crea un formulari amb opcions per al backend.
- **update():** En esta funció és on es guarden les opcions a la base de dades.

Per a afegir i carregar el widget, s'ha de registrar amb la funció `register_widget(classe)` i afegir eixa acció (`add_action()`) amb el hook 'widgets\_init'.

Per últim, però no menys important, un dels fitxers que componen una part fonamental de la vista són les fulles d'estil o més conegudes com els arxius `.css`. Estos fitxers són els que li donen el disseny que necessitem a les nostres plantilles. Permeten canviar els colors, les fonts, les posicions, ordenar elements i moltes coses més. Per defecte existeix un fitxer `style.css` en tots els temes que és on podrem els elements HTML segons l'ID, classe o tipus.

Començarem analitzant els arxius PHP que donen la vista a l'aplicació que es troben en el directori del tema fill (`wp-content/themes/bootscore-child-main`) sense entrar en massa detalls.

### 3.4.1.4 Barra de navegació

Una de les parts més importants de tota pàgina web és la capçalera (**header.php**). En este cas, en la capçalera despleguem la barra de navegació, que ens acompanya en totes les vistes i, a més a més, s'ha implementat de manera fixa per a que mentre ens desplaçem, ja siga entre vistes o cap avall i amunt, estiga sempre present i puguem fer-la servir. Pel que fa al disseny, s'ha escollit alguna cosa senzilla. De dreta a esquerra trobem: En primer lloc el botó que ens porta a la vista d'iniciar sessió. A continuació disposem de les banderes amb els dos idiomes disponibles de la Web (Valencià i Castellà). Després tenim el menú amb els enllaços estàtics a les vistes que té la pàgina i on pot accedir tothom (Inici, Qui Som?, Notícies, Restaurant i Anunciants). Este menú i els seus enllaços s'han configurat des del panel d'administració en Aparència->Menús. Per últim, està el nom de la urbanització a l'esquerra, que funciona com a enllaç per a pujar dalt de la pàgina si estem molt cap avall. El que s'ha descrit anteriorment és la vista per a un ordinador, però com ja s'ha dit, esta pàgina web és adaptable per a tots els dispositius i, per a les "tablets" i els dispositius mòbils tots els enllaços del menú canvien un poc: Es vorà, a la dreta, un botó quadrat amb tres línies horitzontals que, només cliquem, ens desplegarà un menú amb totes les opcions que s'han esmentat, ja que és un menú massa extens, que desquadraria la pàgina web, com per a mostrar-lo de la mateixa manera que es fa amb un ordinador.

S'ha construït amb la funció **wp\_nav\_menu()** que construeix un menú on des del panel d'administració de WordPress després es poden afegir els camps que tindrà. També s'han utilitzat components de Bootstrap [21]

```
<!-- Navigation-->
<nav class="navbar navbar-expand-lg navbar-light fixed-top py-3<?php echo (is_front_page() ? "" : " navbar-shrink"); ?>" id="mainNav">
  <div class="container px-4 px-lg-5">
    <div id="buttonbaradello">
      <a class="navbar-brand" href="#page-top"><?php bloginfo('name'); ?></a>
    </div>
    <button class="navbar-toggler navbar-toggler-right" type="button" data-bs-toggle="collapse" data-bs-target="#navbarResponsive" aria-controls="navbarResponsive"
    <div class="collapse navbar-collapse" id="navbarResponsive">
      <?php
        wp_nav_menu(array(
          'theme_location' => 'main-menu',
          'container' => false,
          'menu_class' => '',
          'fallback_cb' => '__return_false',
          'items_wrap' => '<ul id="bootscore-navbar" class="navbar-nav ms-auto my-lg-0 %2$s"%3$s</ul>',
          'depth' => 2,
          'walker' => new bootstrap_5_wp_nav_menu_walker()
        ));
      ?>
    </div>
  </div>
</nav>
```

La barra de navegació no serà sempre igual. Quan algun usuari inicié sessió, canviarà un poc. Òbviament, el botó d'iniciar sessió ara serà de tancar sessió, però també ens apareixerà un botó desplegable, amb el nom de l'usuari, entre el botó d'iniciar/tancar sessió i les banderes de l'idioma. Este nou botó/enllaç, ens servira per saber si estem en la Intranet de la pàgina web i també, per a poder navegar sobre aquelles vistes que sols disposen els membres de la urbanització (Incidències, Assemblees, Foro, Enquestes i Perfil). Quan l'usuari clique sobre el seu nom, se li desplegaran les opcions anteriors per a navegar. És un format que s'ha elegit per a no canviar massa la barra de navegació i fer la navegació el més intuïtiva possible.

Esta part s'ha construït utilitzant la funció `is_used_logged_in()` per comprovar si l'usuari ha iniciat sessió i, si és el cas, mostrar el menú desplegable de la Intranet. També s'ha utilitzat `wp_get_current_user()->user_login` per obtenir el nom de l'usuari que ha iniciat sessió que es mostrarà en el menú desplegable. S'utilitza `get_site_url()` per obtenir el domini de la pàgina web. La funció `wp_logout_url()` serveix per a que l'usuari tanque sessió quan clique en el botó corresponent.

```
<?php
// user login
if ( is_user_logged_in() ) {
}
<!-- Sesi3n iniciada -->

<!-- <div class="dropdown"> -->
<div class="dropdown" id="buttonuser">
<a class="btn btn-secondary dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-bs-toggle="dropdown" aria-expanded="false">
<?php echo wp_get_current_user()->user_login ?>
</a>

<ul class="dropdown-menu" aria-labelledby="dropdownMenuLink">
<li><a class="dropdown-item" href=" <?php echo get_site_url() . "/perfil" ?> " > Perfil </a></li>
<li><a class="dropdown-item" href=" <?php echo get_site_url() . "/foro" ?> " > Foro </a></li>
<li><a class="dropdown-item" href=" <?php echo get_site_url() . "/assembly" ?> " > Asambleas </a></li>
<li><a class="dropdown-item" href=" <?php echo get_site_url() . "/issue" ?> " > Incidencias </a></li>
<li><a class="dropdown-item" href=" <?php echo get_site_url() . "/encuestas" ?> " > Encuestas </a></li>
</ul>
</div> <!-- </div> -->

<!-- Cerrar sesi3n -->
<div id="cerrarsesion">
<a href = " <?php echo wp_logout_url(); ?>">
<button class="btn btn-primary ms-1 text-white"> Cerrar sesi3n
</button>
</a>
</div>

<?php
// user logout
} else {
}

<!-- Iniciar sesi3n -->
<div id="iniciarsesion">
<a href = "/login">
<button class="btn btn-primary ms-1 text-white">
Iniciar sesi3n
</button>
</a>
</div>

<?php
}
```

A m3s, s'ha utilitzat codi JavaScript per fer la barra fixa quan ens desplacem cap a baix.

### 3.4.1.5 Vista Inici

El fitxer **front-page.php** s'ha construït inspirant-me en la majoria de pàgines d'inici que solen tindre les aplicacions web com és: una capçalera, un xicotet resum de l'associació, un mapa amb la ubicació i un formulari de contacte. De totes maneres, no sols ha sigut una inspiració, perquè principalment s'han tingut en compte les necessitats que tenia l'Associació de veïns com era mostrar les últimes notícies, mostrar la ubicació on es troben i tindre un formulari de contacte per qualsevol que necessite enviar un correu a l'Associació. Doncs, sumant els punts anteriors, es va arribar a un punt comú on es dividiria este fitxer per seccions que serien, ordenadament, les següents: Una capçalera amb una imatge de la urbanització, les últimes notícies i enllaços per a poder accedir a elles, un mapa de google maps amb la ubicació marcada de la urbanització i, per últim, un formulari de contacte que enviara els missatges al correu de l'associació.

L'apartat del formulari és el que més desenvolupament ha tingut. Per una part, al codi HTML s'ha construït un "form post" on els camps són el nom, l'email i el text.

```
<section class="page-section bg-primary" id="contact">
  <div class="container px-4 px-lg-5">
    <div class="row gx-4 gx-lg-5 justify-content-center">
      <div class="col-lg-8 text-center">
        <h2 class="text-white mt-0">Contacto</h2>
        <hr class="divider divider-light" />
        <form method="post">
          <div class="mb-3 text-start">
            <label for="nameform" class="form-label text-white">Nombre</label>
            <input type="text" class="form-control" name="nameform" id="nameform" aria-describedby="nameHelp" />
          </div>
          <div class="mb-3 text-start">
            <label for="email" class="form-label text-white">Email</label>
            <input type="email" class="form-control" name="email" id="email" aria-describedby="emailHelp">
          </div>
          <div class="mb-3 text-start">
            <label for="textarea" class="form-label text-white">Texto</label>
            <textarea class="form-control" name="textarea" id="textarea" rows="3"></textarea>
          </div>
          <!--<a class="btn btn-light btn-xl" href="#services">Enviar</a-->
          <input class="btn btn-light btn-xl" value="Enviar" type="submit" name="form_button"></input>
        </form>
      </div>
    </div>
  </div>
</section>
```

Per l'altra part, el codi PHP recull la petició post del botó amb el nom "form\_button" i les dades. Crea la capçalera i prepara el cos del email. Finalment, s'utilitza la funció **wp\_mail(destinatari, assumpte, cos, capçaleres)** que envia el missatge.

```

<?php
if(isset($_POST['form_button'])) {
    $to = 'baradellogelat@gmail.com';
    $subject = 'Nuevo formulario';

    $name = $_POST['nameform'];
    $email = $_POST['email'];
    $text = $_POST['textarea'];

    $headers = array('Content-Type: text/html; charset=UTF-8');

    $body = $name. ' con email ' . $email. ' ha enviado un formulario desde la web del Baradello Gelat.<br>';
    $body .= 'Con el siguiente mensaje: <br>';
    $body .= $text;

    if ($name && $email && $text) {
        wp_mail($to, $subject, $body, $headers);
        wp_redirect(get_site_url());
        exit();
    }
}
?>

```

#### 3.4.1.6 Vista «Qui som?»

La pàgina de “**¿Quiénes somos?**” s’ha deixat en blanc amb una plantilla estàndard ja que és un apartat que depèn completament del que vulga posar la urbanització de veïns. També, s’havia de tindre en compte que el contingut d’esta pàgina és molt dinàmic i que en qualsevol moment es deu poder canviar de la forma més còmoda. En este cas, la millor manera és des del panel d’administració de WordPress en l’apartat de pàgines, on se li pot donar un format bastant acceptable.

### 3.4.1.7 Vista Notícies

La pàgina de les Notícies està basada en els custom post types de WordPress. Per defecte, existeixen els Posts en WordPress i estos s'utilitzaran per a les notícies. La plantilla on es llisten els posts és **archive.php**, encara que s'han fet unes modificacions per a cobrir les necessitats dels usuaris. En quant a la vista, qualsevol usuari podrà visualitzar les notícies en forma de llista on apareixerà el títol, la data i una breu introducció al text escrit per saber de què tracta. També, entre les notícies i la capçalera, trobarem la part de filtrat de les notícies. Es podran buscar per text, escrivint en la casella que hi ha, i entre dos dates que seran les dos caselles que hi ha. A més a més, cada notícia inclou un botó (Veure/Ver) amb un enllaç que ens porta a la pàgina per llegir la notícia sencera i on es podran visualitzar els fitxers que s'hagen inclòs. Quan els usuaris inicien sessió, aquells usuaris amb permisos especials sobre esta vista (administradors i junta) tindran funcionalitats extra en esta vista. El botó de Veure desapareixerà i es convertirà en un botó Acció amb un menú desplegable amb tres opcions: Vore, Editar i Eliminar les notícies. El botó d'editar ens farà navegar al post corresponent al panel d'administració de WordPress on podrem modificar diferents aspectes de la notícia seleccionada.

Per a mostrar les notícies, s'utilitzaran les funcions **have\_posts()** per comprovar si existeixen i **the\_post()** que invoca la notícia que s'està recorrent en este moment mitjançant un bucle. Després, per mostrar la imatge de la notícia s'invoca la funció **get\_the\_post\_thumbnail\_url()**. També s'utilitza la funció **the\_permalink()** que és l'enllaç a la notícia que estarà sobre el títol de la notícia, invocat amb **the\_title()**. Amb **get\_the\_date()** mostrem la data de publicació de la notícia. Amb **echo substr(get\_the\_excerpt(),0,200)** s'estan mostrant 200 caràcters del resum de la notícia per mostrar-ho baix del títol i donar una introducció.

```
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?>
<div class="row g-5">
  <div class="col card horizontal mb-3 border border-0 py-4" style="min-height: 200px;">
    <div class="row g-0 justify-content-center">
      <div class="col-md-2">
        <?php if (has_post_thumbnail()) : ?>
          
        <?php else : ?>
          
        <?php endif; ?>
      </div>
      <div class="col-md-8">
        <div class="card-body">
          <h3 class="card-title"><?php the_title(); ?></div><div class="card-text"><?php the_excerpt(); ?></div></div>
          <div class="card-text"><div class="notranslate"><?php echo substr(get_the_excerpt(),0,200); ?></div></div>
          <?php if (has_user_role('editor','administrator','board')) : ?>
            <div class="dropdown">
              <button type="button" class="btn btn-secondary dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">
                Acción
              </button>
              <ul class="dropdown-menu">
                <li><a class="dropdown-item" href="#" onclick="window.location='<?php the_permalink(); ?>'>Ver</a></li>
                <li><a class="dropdown-item" href="#">Editar</a></li>
                <li><a class="dropdown-item" href="#">Eliminar</a></li>
              </ul>
            </div>
          <?php else : ?>
            <a class="btn btn-secondary" href="#">Ver</a>
          <?php endif; ?>
        </div>
      </div>
    </div>
  </div>
</div>
<?php endwhile; ?>
<?php else : ?>
  No hay noticias.
<?php endif; ?>
```

La funció **has\_user\_role(['editor','administrator','board'])** comprova si l'usuari que ha iniciat sessió té algun dels rols esmentats i, si és així, mostra el botó de les accions que sols podran realitzar els administradors. Este botó és un desplegable amb: Un enllaç per vore la notícia (**the\_permalink()**), un enllaç per editar la notícia des del panel d'administració web (**edit\_post\_link()**) i un enllaç que executa un mètode post amb l'acció d'eliminar on envia l'ID del post per utilitzar-lo en la funció **wp\_delete\_post()** que elimina la notícia seleccionada.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $post_id = $_REQUEST["post_id"];
    switch ($_REQUEST["action"]) {
        case "delete":
            // Delete post
            wp_delete_post($post_id);
            wp_redirect($_SERVER['REQUEST_URI']);
            exit();
            break;
        default:
    }
}
```



### 3.4.1.8 Vista Restaurant

En el cas de la pàgina del Restaurant, s'ha creat una plantilla, de la manera que s'ha explicat anteriorment, amb el nom de **page-restaurante.php** que permet als administradors, als membres de la junta i als usuaris del restaurant afegir fitxers (que el més normal és que siguin PDF o imatges) a la pàgina. També, els usuaris administradors i els de la junta tenen permisos per a modificar la pàgina en l'apartat de pàgines del panel d'administració. I, finalment, un usuari administrador amb coneixements en programació, podrà dissenyar la plantilla al seu gust amb les necessitats que tinguen.

Amb la funció **has\_user\_role()** comprovem si l'usuari té els permisos per a vore el botó que permet afegir fitxers a la pàgina del Restaurant. Es creen dos funcions en JavaScript, una per seleccionar l'arxiu i l'altra per guardar-lo.

```
<?php
    if (has_user_role(['editor','administrator','restaurant','board'])) :
?>
<button class="btn btn-link" onmousedown="event.preventDefault()" type="button" onclick="selectFile(event)">Añadir fichero</button>
<div id="pnFiles" style="display:none;">
    <form id="formAddAttachment" method="post" enctype="multipart/form-data">
        <input type="hidden" name="post_id" value="<?php the_ID(); ?>" >
        <input type="hidden" name="action" value="add_attachment">
        <input id="file1" name="file1" type="file" onchange="saveFile(event)">
    </form>
</div>

<script>
    function selectFile(event) {
        console.log('selectFile()');
        document.getElementById('file1').click();
    }
    function saveFile(event) {
        console.log('saveFile()');
        let form = document.getElementById('formAddAttachment');
        form.submit();
    }
</script>
<?php endif; ?>
```

Per a mostrar els fitxers carregats, es fa ús de la funció **get\_attached\_media(post\_id)** que obté tots els fitxers adjuntats a un post. Després es recorre la llista de fitxers i van mostrant-se amb les opcions de descarregar i obrir. Sols els usuaris administradors podran Eliminar estos fitxers adjuntats.

```

<?php
$attachments = get_attached_media( '', get_the_ID() );
foreach ( $attachments as $attachment):
    $mime = get_post_mime_type($attachment);
    $thumbnail = wp_get_attachment_image_src ( $attachment->ID, 'thumbnail' );
    $url = wp_get_attachment_url($attachment->ID);
?>
<tr>
<td scope="row" class="align-middle" >
<a href="<?=$url ?>" target="_blank">
<?php if ( $thumbnail): ?>

<?php else: ?>
<i style="font-size: 3em;" class="fa-solid <?=$mime_to_icon($mime) ?>" ></i>
<?php endif; ?>
</a>
</td>
<td class="align-middle" >
<?=$attachment->post_title ?>
</td>
<td class="align-middle">
<div class="dropdown">
<button class="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown" aria-expanded="false">
    Acció
</button>
<ul class="dropdown-menu">
<li><form><a class="dropdown-item" href="<?=$url ?>" target="_blank">Abrir</a></form></li>
<li><form><a class="dropdown-item" href="<?=$url ?>" download>Descargar</a></form></li>
<?php
    if (has_user_role(['editor','administrator','restaurant','board'])) :
?>
<li><form method="post" enctype="multipart/form-data">
<input type="hidden" name="post_id" value="<?php the_ID(); ?>" >
<input type="hidden" name="baradello_attachment_id" value="<?php echo $attachment->ID; ?>" >
<input type="hidden" name="action" value="delete_attachment">
<a class="dropdown-item" href="#" onclick="this.parentNode.submit()">Eliminar</a>
</form></li>
<?php endif; ?>
</ul>
</div>
</td>
</tr>
<?php endforeach; ?>

```

Les funcions d'afegir i eliminar s'invoquen mitjançant un formulari post que depenen de l'acció. Per afegir, s'utilitza la funció **is\_uploaded\_file()** per comprovar si s'estan enviant fitxers i **media\_handle\_upload()** guarda un arxiu enviat des d'una sol·licitud POST i crea una publicació adjunta per ell. Per eliminar, s'obté l'ID del fitxer que es vol eliminar i amb la funció **wp\_delete\_attachment()** s'esborra.

### 3.4.1.9 Vista Anunciants

Per a la pàgina dels Anunciants s'ha creat un custom post type "advertiser" i s'utilitzarà la plantilla **archive-advertiser.php** que es personalitzarà per a canviar-li l'aspecte i diferenciar-la respecte a altres pàgines com, per exemple, la de les Notícies. En este cas, la pàgina d'Anunciants sols tindrà un buscador per a filtrar per text, ja que filtrar per data no tindria massa sentit. Els anunciants apareixeran, en files de tres, en forma de quadrícula, on cada anunciament tindrà la seua imatge, el títol i una breu descripció del que fa. A més a més, qualsevol usuari podrà viatjar a vore la pàgina individual de l'anunciament per llegir més informació. Pel que fa als usuaris amb permisos especials (administradors i junta) tindran un botó d'Afegir Anunciament que els portarà al panel d'administració on podran afegir l'anunciament i les altres característiques. Estos mateixos usuaris podran eliminar i editar individualment els anunciants des de la mateixa pàgina amb el botó Acció que desplegarà un submenú.

Esta plantilla és molt pareguda a la de les Notícies, s'utilitzen les mateixes funcions per mostrar els anunciants: **have\_posts()**, **the\_post()**, **get\_the\_post\_thumbnail\_url()**, **the\_permalink()**, **get\_the\_excerpt()**.

```
<?php if (have_posts()) : ?>
<div class="row row-cols-1 row-cols-md-3 g-4">
  <?php while (have_posts()) : the_post(); ?>
    <div class="col">
      <div class="card">
        <?php if (has_post_thumbnail()) : ?>
          
        <?php else : ?>
          
        <?php endif; ?>
        <div class="card-body">
          <h5 class="card-title"><a style="text-decoration: none; color: gray;" href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h5>
          <p class="card-text"><?php echo substr(get_the_excerpt(),0,200); ?></p>
          <?php if (has_user_role('editor', 'administrator', 'board')): ?>
            <div class="dropdown text-center">
              <button type="button" class="btn btn-secondary dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">
                Acción
              </button>
              <ul class="dropdown-menu">
                <li><a class="dropdown-item" href="#" onclick="window.location='<?php the_permalink(); ?>'">Ver</a></li>
                <li><?php edit_post_link('Editar','','','dropdown-item'); ?></li>
                <li><a class="dropdown-item" href="<?wp_nonce_url( get_site_url() . '/wp-admin/post.php?action=delete&post=' . $post->ID, 'delete-post_' . $post->ID) ?>">Eliminar</a></li>
              </ul>
            </div>
          <?php else : ?>
            <div class="text-center">
              <a class="btn btn-secondary" href="<?php the_permalink(); ?>">Ver</a>
            </div>
          <?php endif; ?>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
```

Per eliminar els anunciants s'utilitza **wp\_delete\_post()**. Per comprovar si l'usuari té permís per esborrar i per vore el botó d'afegir anunciants també s'utilitza **has\_user\_role()**.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $post_id = $_REQUEST["post_id"];
    switch ($_REQUEST["action"]) {
        case "delete":
            // Delete post
            wp_delete_post($post_id);
            wp_redirect($_SERVER['REQUEST_URI']);
            exit();
            break;
        default:
    }
}

global $post;
global $wp_query;

get_header();
?>

<div id="content" class="site-content container">

    <!-- New advertiser button -->
    <?php if (has_user_role(['editor','administrator','board'])) : ?>
        <form method="post" action="<?php site_url() ?>/wp-admin/post-new.php?post_type=advertiser">
            <div class="row gx-md-0 justify-content-end">
                <div class="col-md-2 col-auto">
                    <div class="d-grid gap-2">
                        <button class="btn btn-primary text-white" type="submit">Nuevo anunciante</button>
                    </div>
                </div>
            </div>
        </form>
    <?php endif; ?>
```

### 3.4.1.10 Vista Inici Sessió

La pàgina d'inici de sessió és una pàgina bàsica i plana que recull un usuari i una contrasenya i fa una comprovació sobre la base de dades per veure si les dades són correctes. En el cas que l'autenticació siga satisfactòria l'usuari podrà veure el seu nom d'usuari amb el submenú corresponent en la barra de navegació. Si l'autenticació és errònia, la vista serà la mateixa que abans. Per als casos on l'usuari pugui oblidar la contrasenya, s'ha preparat un enllaç que diu "He oblidat la contrasenya" per a navegar a una altra pàgina molt pareguda a l'anterior però amb una casella per introduir una direcció de correu i enviar un missatge. Este missatge contindrà un enllaç, que ens portarà a una pàgina on introduïrem la nova contrasenya.

Esta vista utilitza un formulari per obtenir el nom d'usuari i contrasenya amb una petició POST. També es comprova si l'usuari ha iniciat sessió, perquè si és el cas, no podrà veure esta pàgina.

```
if (!$logged_in) {
    get_header();
}
<div class="container">
  <div class="row justify-content-center">
    <div class="col-xl-5 col-md-8">
      <form method="post" class="bg-white rounded shadow-5-strong p-5">
        <div class="form-floating mb-4">
          <input type="username" id="username" name="username" class="form-control" placeholder="Nombre de usuario" />
          <label class="form-label" for="username">Nombre de usuario</label>
        </div>
        <div class="form-floating mb-4">
          <input type="password" id="password" name="password" class="form-control" placeholder="Password"/>
          <label for="password">Contraseña</label>
        </div>
        <div class="row mb-4">
          <div class="col text-center">
            <!-- Simple link -->
            <a href="<?php echo get_site_url().'/recuperar-contrasena' ?>">He olvidado la contraseña</a>
          </div>
        </div>
        <!-- Submit button -->
        <button type="submit" class="btn btn-primary btn-block text-white">Entrar</button>
      </form>
    </div>
  </div>
</div>
<?php
```

Si l'usuari ha iniciat sessió, serà redirigit a la pàgina principal per defecte, si no, seguirà veient la pàgina. Per a l'inici de sessió, s'utilitza la funció **wp\_signon()** que autentica l'usuari i retorna la classe WP\_User. La funció **wp\_set\_current\_user()** canvia l'usuari actual per l'ID o el nom i **wp\_set\_auth\_cookie()** estableix les cookies d'autenticació per identificar l'usuari.

```

//user in logged out state
if (is_user_logged_in()) {

    wp_redirect(get_site_url());
    exit();

} else if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    $username = $wpdb->escape($_POST['username']);
    $password = $wpdb->escape($_POST['password']);

    $login_array = array();
    $login_array['user_login'] = $username;
    $login_array['user_password'] = $password;
    $login_array['remember'] = true;

    $verify_user = wp_signon($login_array, false);

    // if the user and password are correct redirect the user to the home page
    if (!is_wp_error($verify_user)) {
        $logged_in = true;
        do_action('wp_login', $verify_user->user_login, $user);
        wp_set_current_user( $verify_user->ID );
        wp_set_auth_cookie( $verify_user->ID );
        //echo "<script>>window.location = '" . get_site_url() . "'</script>";
        if (isset($_REQUEST['redirect'])) {
            wp_redirect($_REQUEST['redirect']);
            exit();
        } else {
            wp_redirect(get_site_url());
            exit();
        }

        // if the user or password are wrong the code shows a message error
    }
}
}

```

Fins ara s'havien mencionat aquelles pàgines que no necessiten d'autenticació per a ser utilitzades, és a dir, que qualsevol persona, siga de l'Associació o no, podrà visualitzar-les. A continuació, van a esmentar-se aquelles pàgines que requereixen de l'inici de sessió amb un usuari i contrasenya.

### 3.4.1.11 Vista Perfil d'Usuari

La primera pàgina important és la del Perfil d'Usuari. Esta pàgina s'ha construït sobre les necessitats principals que podien tindre els usuaris. Per una banda, és important mostrar informació com el nom, cognoms i correu. Esta informació es mostra a l'esquerra baix d'un avatar. Per altra banda, és important donar opció a que l'usuari modifique el seu correu o contrasenya. Açò serà possible en la part del centre i dreta on estan les caselles del correu i de la contrasenya per modificar-ho. Si el correu ja està en ús, no canviarà.

Per a la vista del Perfil, es mostra els camps del nom i el correu i sols es permet canviar la contrasenya i la direcció de correu.

```
echo '<p class="error"> . implode("<br />", $error) . '</p>'; ?>
<div class="container rounded bg-white mt-5 mb-5">
  <div class="row">
    <div class="col-md-3 border-right">
      <div class="d-flex flex-column align-items-center text-center p-3 py-5">
      <div class="p-3 py-5">
        <div class="d-flex justify-content-between align-items-center mb-3">
          <h4 class="text-center">Perfil de Usuario</h4>
        </div>
        <form method="post" id="adduser" action="<?php the_permalink(); ?>">
          <div class="row mt-3">
            <div class="col-md-12">
              <p class="form-email">
                <div class="parametros">
                  <div><label for="email"><?php _e('Email: ', 'profile'); ?></label> </div>
                  <div><input class="text-input text-center" name="email" type="text" id="email" value="<?php the_author_meta('user_email', $current_user->ID); ?>" /> </div>
                </div>
              </p><!-- .form-email -->
            </div>
            <div class="col-md-12">
              <p class="form-password">
                <div class="parametros">
                  <div><label for="pass1"><?php _e('Nueva contraseña: ', 'profile'); ?></label> </div>
                  <div><input class="text-input" name="pass1" type="password" id="pass1" /> </div>
                </div>
              </p><!-- .form-password -->
            </div>
            <div class="col-md-12">
              <p class="form-password">
                <div class="parametros">
                  <div><label for="pass2"><?php _e('Repita la contraseña: ', 'profile'); ?></label> </div>
                  <div><input class="text-input" name="pass2" type="password" id="pass2" /> </div>
                </div>
              </p><!-- .form-password -->
            </div>
          </div>
          <?php
          //action hook for plugin and extra fields
          //do_action('edit_user_profile', $current_user);
          ?>
          <p class="form-submit">
            <?php //echo $referer; ?>
            <div class="mt-5 text-center"><input name="updateuser" type="submit" id="updateuser" class="submit button btn btn-primary profile-button" value="<?php _e('Guardar', 'profile'); ?>"
            <?php wp_nonce_field('update-user') ?>
            <input name="action" type="hidden" id="action" value="update-user" />
          </p><!-- .form-submit -->
        </div>
      </div>
    </div>
  </div>
</div>
```

Mitjançant codi PHP es comproven els canvis que s'han guardat i s'apliquen contra la base de dades. La direcció de correu es comprova abans per si ja existeix.

```

if ('POST' == $_SERVER['REQUEST_METHOD'] && !empty($_POST['action']) && $_POST['action'] == 'update-user') {
    /* Update user password. */
    if (!empty($_POST['pass1']) && !empty($_POST['pass2'])) {
        if ($_POST['pass1'] == $_POST['pass2'])
            wp_update_user(array('ID' => $current_user->ID, 'user_pass' => esc_attr($_POST['pass1')));
        else
            $error[] = __('Las contraseñas no coinciden. La contraseña no ha sido actualizada.', 'profile');
    }

    /* Update user information. */
    if (!empty($_POST['url']))
        wp_update_user(array('ID' => $current_user->ID, 'user_url' => esc_url($_POST['url'])));
    if (!empty($_POST['email'])) {
        if (!is_email(esc_attr($_POST['email'])))
            $error[] = __('El Email introducido no es válido. Prueba otra vez.', 'profile');
        /*elseif (email_exists(esc_attr($_POST['email'])))
            $error[] = __('Este Email ya está en uso. Prueba otra vez.', 'profile');*/
        else {
            wp_update_user(array('ID' => $current_user->ID, 'user_email' => esc_attr($_POST['email'])));
        }
    }

    if (!empty($_POST['first-name']))
        update_user_meta($current_user->ID, 'first_name', esc_attr($_POST['first-name']));
    if (!empty($_POST['last-name']))
        update_user_meta($current_user->ID, 'last_name', esc_attr($_POST['last-name']));
    if (!empty($_POST['description']))
        update_user_meta($current_user->ID, 'description', esc_attr($_POST['description']));

    /* Redirect so the page will show updated info.*/
    /*I am not Author of this Code- I don't know why but it worked for me after changing below line to if ( count($error) == 0){ */
    if (count($error) == 0) {
        //action hook for plugins and extra fields saving
        do_action('edit_user_profile_update', $current_user->ID);
        wp_redirect(get_permalink());
        exit;
    }
}
}

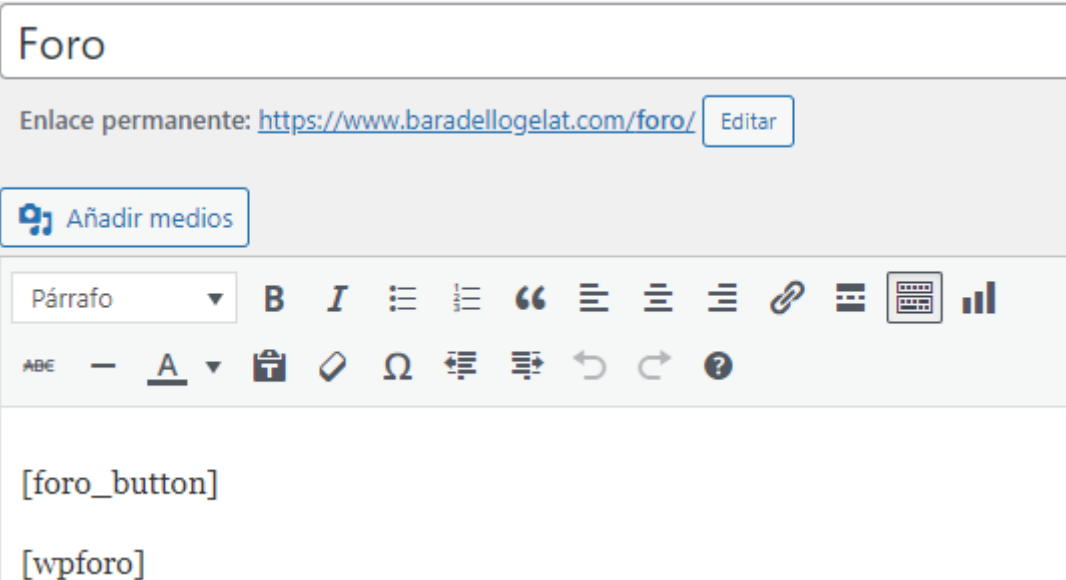
```



### 3.4.1.12 Vista Foro

El foro s'ha construït sobre una pàgina creada en el panel d'administració de WordPress. En la pàgina s'han afegit dos shortcodes. Els shortcodes són cadenes de text entre claudàtors que criden a una funció que mostrarà un frontend totalment funcional. El primer crida a un botó, que sols li apareixerà als administradors o els membres de la junta, que servirà per a gestionar el foro ja que ens portarà a la pàgina de configuració del plugin. El segon és el que crida a la pàgina del foro, totalment configurat en el plugin. És un shortcode molt complet que mostra la interfície de les categories, els temes i tots els comentaris... També ofereix vistes com la del perfil d'usuari o altres que es detallaran en l'apartat del plugin.

A continuació es mostren els dos shortcodes en la pàgina:



The screenshot shows a WordPress page editor for a page titled "Foro". At the top, there is a text area with the permanent link "https://www.baradellogelat.com/foro/" and an "Editar" button. Below this is a "Añadir medios" button. A rich text editor toolbar is visible, containing various icons for text formatting (bold, italic, list, quote, indent, outdent, link, unlink, table, columns) and other functions (undo, redo, help). Below the toolbar, the page content area contains two shortcodes: `[foro_button]` and `[wpforo]`.

El shortcode “foro\_button” s’ha creat en funcions.php. Sols li apareixerà als administradors i és un botó que condueix al panel d’administració del foro.

```
function foro_button()
{
    if (has_user_role(['editor', 'administrator', 'board'])) {
        return '<!-- New foro button -->

        <form method="post" action="' . get_site_url() . '/wp-admin/admin.php?page=wpforo-forums">
        <div class="row gx-md-0 justify-content-end">
            <div class="col-md-2 col-auto">
                <div class="d-grid gap-2">
                    <button class="btn btn-primary text-white" type="submit">Gestionar el foro</button>
                </div>
            </div>
        </div>
        </form>';
    } else {
        return '';
    }
}

add_shortcode('foro_button', 'foro_button');
```

### 3.4.1.13 Vista Assemblees

Per a la pàgina de les Assemblees s'ha creat un custom post type "assembly" i s'utilitzarà la plantilla **archive-assembly.php** per a mostrar-les. Per a que els usuaris administradors i de la junta puguin afegir assemblees fàcilment, tenen un botó per a navegar al panel d'administració web de les assemblees. Qualsevol usuari disposarà d'un apartat de filtrat, com en anteriors vistes, on es podrà buscar assemblees compreses entre dos dates i pel seu títol. Les assemblees es mostraran en forma de taula amb: el nom, la data i l'acció que es pot realitzar sobre ella. Els usuaris normals de la urbanització sols podran vore les assemblees en una vista individual per a llegir el text que s'haja publicat i per vore les imatges o fitxers adjunts. No obstant, els membres de la junta o administradors també podran editar-les o eliminar-les.

Es mostren en forma de llista i utilitzen les mateixes funcions que els altres tipus de posts.

```
<!-- Assembly list -->
<table class="table table-hover table-responsive-sm">
  <thead>
    <tr>
      <th scope="col">Asamblea </th>
      <th scope="col">Fecha </th>
      <th scope="col">Acción </th>
    </tr>
  </thead>
  <tbody class="table-group-dividen">
    <?php
    if (have_posts()):
      while(have_posts()):
        the_post();
      >
      <tr>
        <td><a class="link-dark" href="<?php the_permalink(); ?>"><?php echo the_title(); ?> </a> </td>
        <td><?php the_time('G:i, j F Y') ?> </td>
        <td>
          <?php if (has_user_role(['editor','administrator','board'])) : ?>
          <div class="btn-group">
            <button type="link" class="btn" onclick="window.location=<?php the_permalink(); ?>">Ver </button>
            <button type="button" class="btn dropdown-toggle dropdown-toggle-split" data-bs-toggle="dropdown" aria-expanded="false">
              <span class="visually-hidden">Toggle Dropdown </span>
            </button>
            <ul class="dropdown-menu">
              <li><?php edit_post_link('Editar','','', 'dropdown-item') ?></li>
              <li><a class="dropdown-item" href="<?php wp_nonce_url( get_site_url() . '/wp-admin/post.php?action=delete&post=' . $post->ID, 'delete-post' . $post->ID) ?>">Eliminar </a></li>
            </ul>
          </div>
          </td>
        </tr>
      <?php endif; ?>
    </tbody>
  </table>
  <?php
  //loop content (template tags, html, etc)
  endwhile;
endfor;
endfor;
```

### 3.4.1.14 Vista Incidències

La pàgina de les incidències també s’ha construït sobre un custom post type anomenat “issue” i que utilitzarà la plantilla **archive-issue.php**. Esta, possiblement siga de les plantilles, dels tipus de posts, més completes i amb més necessitats. Per començar, qualsevol usuari podrà afegir una nova incidència, hi ha un botó que desplega un formulari on es podrà escriure el títol, un text i arxius com poden ser imatges o pdf. Els usuaris normals sols podran veure les incidències que publiquen ells mateixa, per evitar sobrecarregar la pàgina i perquè no té massa sentit que els usuaris veguen les que van publicant altres veïns. Les incidències es visualitzaran en forma de taula amb: El nom, la data, l’estat (proposta, acceptada, rebutjada o tancada) i l’acció que, en este cas, serà de visualitzar-la individualment per a conèixer més detalls i veure els arxius adjunts. Qualsevol usuari també podrà filtrar les incidències segons text, estat i dos dates. Els usuaris amb rol d’administrador i de mantenidor tindran una vista un poc diferent. Primer que tot, podran veure totes les incidències publicades i voran el seu autor en la taula. A més a més, tindran permís per a canviar-li l’estat, segons convinga, en l’apartat de l’acció. Quan un usuari publica una incidència, per defecte, està Proposta. Després, si el mantenidor creu que pot solucionar-ho o va a fer-ho, hauria d’acceptar-la per a que passe a l’estat Acceptada. En este punt, poden passar dos coses. La primera és que el mantenidor solucione la incidència, aleshores deuria tancar-la per a que passe a l’estat Tancada. La segona és que el mantenidor no puga arreglar-ho, per tant, deuria rebutjar-la i la incidència tornaria a estar Proposta a l’espera que algú la torne a Acceptar.

Es crea un botó per afegir una nova incidència que desplega un formulari amb un “modal-dialog” que mostra un camp per al títol, un camp per al cos de la incidència amb **wp\_editor()** (funció que mostra l’editor de text que apareix al afegir qualsevol tipus de post en el panel d’administració de WordPress) i un enllaç per afegir fitxers.

```
<div class="row gx-md-0 justify-content-end">
  <div class="col-md-2 col-auto">
    <div class="d-grid gap-2">
      <button class="btn btn-primary text-white" type="button" data-bs-toggle="modal" data-bs-target="#dlgAddIssue">
        Nueva incidencia
      </button>
    </div>
  </div>
</div>

<form id="formAddIssue" method="post" enctype="multipart/form-data">
  <input type="hidden" name="action" value="new" >
  <div id="dlgAddIssue" class="modal fade" tabindex="-1">
    <div class="modal-dialog modal-lg modal-fullscreen-md-down">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title">Nueva incidencia</h5>
          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
        </div>
        <div class="modal-body">
          <div class="row">
            <div class="col-12 form-floating mb-4">
              <input type="text" id="post_title" name="post_title" class="form-control" placeholder="Título" />
              <label class="form-label" for="username">Título</label>
            </div>
            <div class="col-12">
              <div class="form-text">
                <?php wp_editor("", "post_content", array('media_buttons'=>false)); >
              </div>
              <div class="form-text">
                <button class="btn btn-link" onmousedown="event.preventDefault()" type="button" onclick="selectFile(event)">Añadir fichero</button>
              </div>
              <div id="pnFiles" style="display:none">
                <input id="file1" name="file1" type="file" onchange="saveFile('1', event)">
              </div>
              <ul id="lstFiles" class="list-group list-group-flush"></ul>
            </div>
          </div>
          <div class="modal-footer">
            <button type="button" type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
            <button type="submit" type="button" class="btn btn-primary text-white">Guardar</button>
          </div>
        </div>
      </div>
    </div>
  </form>
</script>
```

S'obtenen les dades i s'utilitza la funció **wp\_insert\_post()** per afegir la incidència i **media\_handle\_upload()** per adjuntar els fitxers. També està l'opció d'enviar notificació per correu quan es creen incidències.

```
case "new":
    $post_title = $_POST['post_title'];
    $post_content = $_POST['post_content'];
    if ($post_title) {
        $new_post = array(
            // 'ID' => '',
            'post_author' => get_current_user_id(),
            'post_content' => $post_content,
            'post_title' => $post_title,
            'post_status' => 'publish',
            'post_type' => 'issue',
            /* 'meta_input' => array(
                'status', 'Propuesta'
            ) */
        );

        $post_id = wp_insert_post($new_post);
        add_post_meta($post_id, 'status', 'Propuesta', true);

        // Add files as attachments
        require_once( ABSPATH . 'wp-admin/includes/image.php' );
        require_once( ABSPATH . 'wp-admin/includes/file.php' );
        require_once( ABSPATH . 'wp-admin/includes/media.php' );

        foreach ($FILES as $name => $file) {
            if (is_uploaded_file($FILES[$name]['tmp_name'])) {
                media_handle_upload($name, $post_id);
            }
        }

        // Notify
        $post = get_post($post_id);
        $ctxt = array(
            'author' => $author,
            'post' => $post
        );
        send_notification(
            $recipients,
            'Nueva incidencia Baradello Gelat',
            'email-issue-new.php',
            $ctxt
        );

        #wp_redirect($redirect);
        echo "<script>>window.location = '" . get_site_url() . '/issue' . "'</script>";
        exit();
    }
    break;
```

El filtrat d'incidències es fa per estat, per dates i per text a buscar. Es construeix sobre un formulari GET.

```

<!-- Search Form -->
<form method="get" action="{?php site_url(); }?issue">
<!--input type="hidden" name="post_type" value="assembly"-->
<div class="row gx-md-0 gx-5 py-5">
<div class="col-md-4">
<input type="text" name="search" class="form-control search-sit" placeholder="Texto a buscar" value="{?= isset($_REQUEST['search'])? esc_html($_REQUEST['search']): '' }?>
</div>
<div class="col-md-2">
<select class="form-select" aria-label="Estado de la incidencia" name="status">
<option <?= ((isset($_REQUEST['status']) or $_REQUEST['status']) == 'Pendiente')? "selected": "" ?> value="Pendiente">Pendiente</option>
<option <?= (isset($_REQUEST['status']) && $_REQUEST['status'] == 'Todas')? "selected": "" ?> value="Todas">Todas</option>
<option <?= (isset($_REQUEST['status']) && $_REQUEST['status'] == 'Cerrada')? "selected": "" ?> value="Cerrada">Cerrada</option>
<option <?= (isset($_REQUEST['status']) && $_REQUEST['status'] == 'Propuesta')? "selected": "" ?> value="Propuesta">Propuesta</option>
<option <?= (isset($_REQUEST['status']) && $_REQUEST['status'] == 'Aceptada')? "selected": "" ?> value="Aceptada">Aceptada</option>
<option <?= (isset($_REQUEST['status']) && $_REQUEST['status'] == 'Rechazada')? "selected": "" ?> value="Rechazada">Rechazada</option>
</select>
</div>
<div class="col-md-2">
<div class="input-group date" data-provide="datepicker">
<input name="from" type="date" class="form-control" value="{?= isset($_REQUEST['from'])? $_REQUEST['from']: '' }?>
<div class="input-group-addon">
<span class="glyphicon glyphicon-th"></span>
</div>
</div>
</div>
<div class="col-md-2">
<div class="input-group date" data-provide="datepicker">
<input name="to" type="date" class="form-control" value="{?= isset($_REQUEST['to'])? $_REQUEST['to']: '' }?>
<div class="input-group-addon">
<span class="glyphicon glyphicon-th"></span>
</div>
</div>
</div>
<div class="col-md-2 col-auto">
<div class="d-grid gap-2">
<button class="btn btn-primary text-white" type="submit"> Buscar </button>
</div>
</div>
</div>
</form>

```

Es mostren les incidències en forma de taula i sols els administradors i membres de manteniment tenen permisos per canviar els estats.

```

<!-- Issue list -->
<table class="table table-hover table-responsive-sm">
<thead>
<tr>
<th scope="col">Incidencia</th>
<th scope="col">Autor</th>
<th scope="col">Fecha</th>
<th scope="col">Estado</th>
<?php if (has_user_role(['editor','administrator','maintenance'])) : ?>
<th scope="col">Acción</th>
<?php endif; ?>
</tr>
</thead>
<tbody class="table-group-divider">
<?php
if (have_posts()):
while(have_posts()):
the_post();
$status = get_post_meta(get_the_ID(), 'status', true);
switch ($status) {
case "Propuesta":
$color = "text-danger";
break;
case "Aceptada":
$color = "text-primary";
break;
case "Rechazada":
$color = "";
break;
case "Cerrada":
$color = "text-success";
break;
default:
$color = "";
}
?>

```

```

<tr>
<td> <a class="link-dark" href="{php the_permalink(); ?}"> <div class=notranslate> {php echo the_title(); ?} </div> </a> </td>
<td> {php the_author_meta('nickname', get_post_field('post_author'))}; ?} </td>
<td> {php the_time('G:i, j F Y')} ?} </td>
<td class="{?=$color ?}"> {php echo get_post_meta(get_the_ID(), 'status', true); ?} </td>
{php if (has_user_role('editor', 'administrator', 'maintenance')) : ?}
<td>
<div class="btn-group">
<button type="link" class="btn" onclick="window.location='{php the_permalink(); ?}'">Ver</button>
<button type="button" class="btn dropdown-toggle dropdown-toggle-split" data-bs-toggle="dropdown" aria-expanded="false"
<span class="visually-hidden">Toggle Dropdown</span>
</button>
<ul class="dropdown-menu">
<li><form><php edit_post_link('Editar', '', '', 'dropdown-item'); ?></form></li>
{php if (get_post_meta(get_the_ID(), 'status', true) != 'Propuesta') : ?}
<li><form method="post">
<input type="hidden" name="post_id" value="{php echo get_the_ID(); ?}">
<input type="hidden" name="action" value="propose">
<a class="dropdown-item" href="#" onclick="this.parentNode.submit()">Proponer</a>
</form></li>
{php endif; ?}
{php if (get_post_meta(get_the_ID(), 'status', true) != 'Aceptada') : ?}
<li><form method="post">
<input type="hidden" name="post_id" value="{php echo get_the_ID(); ?}">
<input type="hidden" name="action" value="accept">
<a class="dropdown-item" href="#" onclick="this.parentNode.submit()">Aceptar</a>
</form></li>
{php endif; ?}
{php if (get_post_meta(get_the_ID(), 'status', true) != 'Rechazada') : ?}
<li><form method="post">
<input type="hidden" name="post_id" value="{php echo get_the_ID(); ?}">
<input type="hidden" name="action" value="reject">
<a class="dropdown-item" href="#" onclick="this.parentNode.submit()">Rechazar</a>
</form></li>
{php endif; ?}
{php if (get_post_meta(get_the_ID(), 'status', true) != 'Cerrada') : ?}
<li><form method="post">
<input type="hidden" name="post_id" value="{php echo get_the_ID(); ?}">
<input type="hidden" name="action" value="close">
<a class="dropdown-item" href="#" onclick="this.parentNode.submit()">Cerrar</a>
</form></li>
{php endif; ?}
<li><form method="post">
<input type="hidden" name="post_id" value="{php echo get_the_ID(); ?}">
<input type="hidden" name="action" value="delete">
<a class="dropdown-item" href="#" onclick="this.parentNode.submit()">Eliminar</a>
</form></li>
</ul>
</div>
</td>

```

Segons l'acció i el valor de l'acció, es fa una petició POST que es recull amb codi PHP i canvia l'estat de la incidència. S'utilitza la funció `update_post_meta()` per canviar l'estat.

```

case "propose":
    update_post_meta( $post_id, 'status', 'Propuesta' );
    // Notify
    $post = get_post($post_id);
    $ctxt = array(
        'author' => $author,
        'post' => $post
    );
    send_notification(
        $recipients,
        'Actualización incidencia Baradello Gelat',
        'email-issue-update.php',
        $ctxt
    );
    #wp_redirect($redirect);
    echo "<script>window.location = '' . get_site_url(). '/issue' . ''</script>";
    exit();
    break;
case "accept":
    update_post_meta( $post_id, 'status', 'Aceptada' );
    // Notify
    $post = get_post($post_id);
    $ctxt = array(
        'author' => $author,
        'post' => $post
    );
    send_notification(
        $recipients,
        'Actualización incidencia Baradello Gelat',
        'email-issue-update.php',
        $ctxt
    );
    #wp_redirect($redirect);
    echo "<script>window.location = '' . get_site_url(). '/issue' . ''</script>";
    exit();
    break;
case "reject":
    update_post_meta( $post_id, 'status', 'Rechazada' );
    // Notify
    $post = get_post($post_id);
    $ctxt = array(
        'author' => $author,
        'post' => $post
    );
    send_notification(
        $recipients,
        'Actualización incidencia Baradello Gelat',
        'email-issue-update.php',
        $ctxt
    );
    #wp_redirect($redirect);
    echo "<script>window.location = '' . get_site_url(). '/issue' . ''</script>";
    exit();
    break;
case "close":
    update_post_meta( $post_id, 'status', 'Cerrada' );
    // Notify
    $post = get_post($post_id);
    $ctxt = array(
        'author' => $author,
        'post' => $post
    );
    send_notification(
        $recipients,
        'Actualización incidencia Baradello Gelat',
        'email-issue-update.php',
        $ctxt
    );
    #wp_redirect($redirect);
    echo "<script>window.location = '' . get_site_url(). '/issue' . ''</script>";
    exit();
    break;

```

### 3.4.1.15 Vista Enquestes

La pàgina de les enquestes s'ha construït preparant un shortcode en el plugin del Foro i cridant-lo des d'una pàgina creada en el panel de configuració de WordPress. També s'ha habilitat un botó per a que administradors i membres de la junta puguin accedir directament a l'apartat on s'afegeixen les enquestes. Podran afegir la pregunta personalitzada amb cert format (tipus de lletres, espais, estructura...), podran afegir tantes respostes com vulguen, podran seleccionar el número de respostes que pot escollir l'usuari i podran escollir la data d'inici i fi de publicació de l'enquesta. Una vegada publicada l'enquesta, apareixerà com a Pendent per als usuaris que no han votat i tindran el botó que els portarà a la pàgina de votació. Una vegada es vote, la pàgina es recarregarà i no donarà opció a votar, indicant que ja s'ha fet. En la pàgina de les enquestes, apareixerà com a Oberta perquè l'usuari ja ha votat però encara no s'ha acabat el plaç i no es poden consultar els resultats. En el moment en que es tanque, apareixerà com a Tancada i qualsevol veí podrà consultar els resultats.

El shortcode obté les enquestes tancades, les obertes i totes. Hi ha una comprovació on la primera condició mostra l'enquesta per poder votar si està oberta i l'usuari no ha votat. La segona condició mostra un missatge de que l'usuari ja ha votat si encara està oberta. I la tercera condició mostra el resultat si l'enquesta està tancada.

```
add_shortcode('get_polls_page','get_polls_page');
function get_polls_page() {
    if (is_user_logged_in()){
        return '';
    }
    global $wpdb;

    # Query to close polls when pollq_expiry < current_time
    $close_polls = $wpdb->query("UPDATE $wpdb->pollsq SET pollq_active = 0 WHERE pollq_expiry < '".current_time('timestamp')."' AND pollq_expiry != 0 AND pollq_active != 0");
    $polls = array();
    $polls = $wpdb->get_results( $wpdb->prepare( "SELECT pollq_id,pollq_question,pollq_expiry,pollq_active FROM $wpdb->pollsq ORDER BY pollq_id DESC;" ) );
    $active_polls = $wpdb->get_col( $wpdb->prepare( "SELECT pollq_id FROM $wpdb->pollsq WHERE pollq_active = 1" ) );

    $id = $_GET['id'];
    if($id){
        $content = '<div class="site-content container">';
        $content .= '
        <nav aria-label="breadcrumb">
        <ol class="breadcrumb">
            <li class="breadcrumb-item active" aria-current="page">Àrea privada</li>
            <li class="breadcrumb-item active"><a href="'.site_url(). '/encuestas">Encuestas</a></li>
        </ol>
        </nav>';
        if ( in_array($id, $active_polls) && can_vote($id) ) {
            $content .= '<div class="notranslate">';
            $content .= display_pollvote($id,$display_loading = true);
            $content .= '</div>';
        } else if ( in_array($id, $active_polls) && !can_vote($id) ) {
            $content .= '<h4> Ya has votado en esta encuesta </h4>';
        } else if ( !in_array($id, $active_polls) ) {
            $content .= '<div class="notranslate">';
            $content .= display_pollresult($id);
            $content .= '</div>';
        }
        $content .= '</div>';
        return $content;
    }
    $content = '<div class="site-content container">';
}
```

S'utilitza la funció **has\_user\_role()** per a que sols els administradors puguin vore el botó d'afegir incidències. Les incidències es mostren en forma de taula amb els camps ID, Enquesta, Estat i Acció.



```

if (has_user_role(['editor', 'administrator', 'board'])) {
$content .=
'<form method="post" action="'.site_url().'/wp-admin/admin.php?page=wp-polls%2Fpolls-add.php">
<div class="row gx-md-0 justify-content-end">
<div class="col-md-2 col-auto">
<div class="d-grid gap-2">
<button class="btn btn-primary text-white" type="submit"> Nueva encuesta </button>
</div>
</div>
</div>
</form>;
}

$content .=
'<table class="table table-hover table-responsive-sm">
<thead>
<tr>
<th scope="col"> ID </th>
<th scope="col"> Encuesta </th>
<th scope="col"> Estado </th>
<th scope="col"> Acción </th>;

```

Amb un bucle for es recorren les incidències. Si l'usuari és administrador pot editar-la. La primera condició mostra el botó de votar si l'usuari encara no ho ha fet i està oberta.

```

foreach($polls as $poll){
$aux = substr($poll->poll_question,0,80);
$str2 = explode(' ', $aux);
$str = explode('<', $str2[0]);
$content .=
'<tr>
<td><a class="link-dark" href="'.site_url().'/encuestas/?id='.$poll->poll_id.'">'.$poll->poll_id.</a> </td>
<td><div class="notranslate">'. $str[0]. '</div></td>;
//if the user can vote and the poll is active
if ( $poll->poll_active == 1 && can_vote($poll->poll_id) ) {
$content .= '<td> <div style="color:blue"> Pendiente </div> </td>;
if (has_user_role(['editor', 'administrator', 'board'])) {
$content .=
'<td>
<div class="btn-group">
<a type="button" class="btn btn-light" href="'.site_url().'/encuestas/?id='.$poll->poll_id.'">Votar</a>
<button type="button" class="btn dropdown-toggle dropdown-toggle-split" data-bs-toggle="dropdown" aria-expanded="false">
<span class="visually-hidden">Toggle Dropdown</span>
</button>
<ul class="dropdown-menu">
<li><a class="dropdown-item" href="'.site_url().'/wp-admin/admin.php?page=wp-polls%2Fpolls-manager.php&mode=edit&id='.$poll->poll_id.'"> Editar </a> </li>
</ul>
</div>
</td>
';
} else {
$content .=
'<td>
<div class="btn-group">
<a type="button" class="btn btn-light" href="'.site_url().'/encuestas/?id='.$poll->poll_id.'">Votar</a>
</div>
</td>
';
}
}
}

```

Si l'usuari ja ha votat però encara està oberta, apareixerà un missatge d'Espera.

```

// if the user can't vote and the poll is active
} else if ( $poll->pollq_active == 1 && !can_vote($poll->pollq_id) ) {
    $content .= ' <td> <div style="color:green"> Activa </div> </td>';
    if (has_user_role(['editor','administrator','board'])) {
        $content .=
        .
        <td>
        <div class="btn-group">
        <a type="button" class="btn btn-light" class="dropdown-item"> Esperar </a>
        <button type="button" class="btn dropdown-toggle dropdown-toggle-split" data-bs-toggle="dropdown" aria-expanded="false">
        <span class="visually-hidden"> Toggle Dropdown </span>
        </button>
        <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="'.site_url().' /wp-admin/admin.php?page=wp-polls%2Fpolls-manager.php&mode=edit&id='.$poll->pollq_id.'"> Editar </a> </li>
        </ul>
        </div>
        </td>
        ;
    } else {
        $content .=
        .
        <td>
        <div class="btn-group">
        <a type="button" class="btn btn-light" class="dropdown-item"> Esperar </a>
        </div>
        </td>
        ;
    }
}

```

Si l'enquesta ja no està activa, es mostrarà un botó per vore la incidència.

```

// if the poll is not active
} else if ( $poll->pollq_active == 0 ) {
    $content .= ' <td> <div style="color:red"> Cerrada </div> </td>';
    if (has_user_role(['editor','administrator','board'])) {
        $content .=
        .
        <td>
        <div class="btn-group">
        <a type="button" class="btn btn-light" class="dropdown-item" href="'.site_url().' /encuestas/?id='.$poll->pollq_id.'"> Ver </a>
        <button type="button" class="btn dropdown-toggle dropdown-toggle-split" data-bs-toggle="dropdown" aria-expanded="false">
        <span class="visually-hidden"> Toggle Dropdown </span>
        </button>
        <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="'.site_url().' /wp-admin/admin.php?page=wp-polls%2Fpolls-manager.php&mode=edit&id='.$poll->pollq_id.'"> Editar </a> </li>
        </ul>
        </div>
        </td>
        ;
    } else {
        $content .=
        .
        <td>
        <div class="btn-group">
        <a type="button" class="btn btn-light" class="dropdown-item" href="'.site_url().' /encuestas/?id='.$poll->pollq_id.'"> Ver </a>
        </div>
        </td>
        ;
    }
}

```

### 3.4.1.16 Plugins

Els plugins en WordPress són un conjunt de codi que afegixen funcionalitats concretes al lloc web i que es poden activar i desactivar segons convinga. Existeixen plugins de pagament i plugins gratuïts. També, qualsevol desenvolupador pot crear el seu propi plugin.

Un dels motius pels quals esta pàgina web s'ha desenvolupat amb WordPress és l'existència d'alguns plugins gratuïts que implementaven certes funcionalitats complexes i han ajudat a agilitzar el projecte.

En la creació d'este lloc web, s'han utilitzat diferents plugins per afegir les funcionalitats requerides per la urbanització.

Per una banda, estan els plugins de codi lliure que s'han utilitzat:

- **WP Mail SMTP.** SMTP (Simple Mail Transfer Protocol) és un protocol de xarxa estàndar per enviar i rebre correus electrònics. Este plugin serveix per a configurar un proveïdor SMTP per enviar els correus correctament. Els clients de correu electrònic lluiten contra l'spam i comproven que els correus s'originen d'on venen. Si per alguna raó no tenen l'autenticació correcta és molt possible que el missatge no arribe al destinatari. A més, està el problema de que la majoria de VPS o els host de WordPress no estan configurats per enviar correus electrònics. En este plugin s'ha de configurar el correu electrònic del remitent (el correu de l'Associació), el nom del remitent i el servici que s'utilitza (Google/Gmail). En la part del client Gmail s'ha de crear un nou projecte en l'API generant un ID de Client i una clau secreta per a que el WordPress s'autentique contra l'API de Google/Gmail i pugui enviar correus. S'ha escollit este plugin per la facilitat de configuració, perquè és gratuït i pel seu bon funcionament.
- **G Translate.** Este plugin és el que farem servir per a les traduccions valencià-castellà. GTranslate utilitza la traducció automàtica del Google Translate, per tant té disponibles molts idiomes. En la configuració d'este plugin es pot establir l'apariència del widget, l'idioma per defecte, el menú on es mostrarà, el tamany de la bandera, entre altres.

## Opciones del widget

Apariencia del widget:	<input type="text" value="Banderas"/>
Traducir de:	<input type="text" value="Español"/>
* Estructura de URL de sub-directorio: <code>http://example.com/ru/</code>	<input type="checkbox"/> <a href="#">* solo para planes de pago</a>
* Estructura de URL de subdominio: <code>http://es.example.com/</code>	<input type="checkbox"/> <a href="#">* solo para planes de pago</a>
Cambiar automáticamente al idioma del navegador:	<input type="checkbox"/>
Mostrar en el menú:	<input type="text" value="Main menu"/>
Mostrar el selector de idioma flotante:	<input type="text" value="No"/>
Mostrar los nombres de idioma nativos:	<input type="checkbox"/>
Tamaño de la bandera:	<input type="text" value="24px"/>
Estilo de la bandera:	<input type="text" value="3D (.png)"/>

També es pot vore la vista prèvia del widget.

## Vista previa del widget



A més, es poden afegir tantes banderes al menú, com idiomes volem traduir:

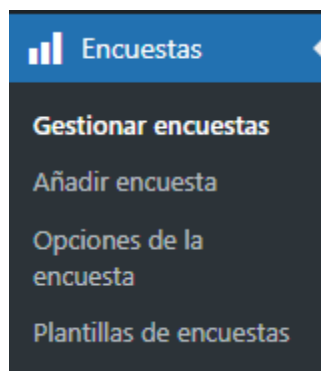
Idiomas de la bandera: [Marcar todo](#) | [Desmarcar todo](#)

HINT: To reorder the languages simply drag and drop them in the list below.

<input type="checkbox"/> Africano	<input type="checkbox"/> Francés	<input type="checkbox"/> Kirguís	<input type="checkbox"/> Serbio
<input type="checkbox"/> Albano	<input type="checkbox"/> Frisón	<input type="checkbox"/> Lao	<input type="checkbox"/> Sesotho
<input type="checkbox"/> Amharca	<input type="checkbox"/> Gallego	<input type="checkbox"/> Latín	<input type="checkbox"/> Shona
<input type="checkbox"/> Àrabe	<input type="checkbox"/> Georgiano	<input type="checkbox"/> Letón	<input type="checkbox"/> Sindhi
<input type="checkbox"/> Armenio	<input type="checkbox"/> Alemán	<input type="checkbox"/> Lituano	<input type="checkbox"/> Cingalés
<input type="checkbox"/> Azerbaiyano	<input type="checkbox"/> Griego	<input type="checkbox"/> Luxemburgo	<input type="checkbox"/> Eslovaco
<input type="checkbox"/> Vasco	<input type="checkbox"/> Gujarati	<input type="checkbox"/> Macedonio	<input type="checkbox"/> Esloveno
<input type="checkbox"/> Bielorruso	<input type="checkbox"/> Creole haitiano	<input type="checkbox"/> Madagascar	<input type="checkbox"/> Somalí
<input type="checkbox"/> Bengalí	<input type="checkbox"/> Hausa	<input type="checkbox"/> Malayo	<input type="checkbox"/> Español
<input type="checkbox"/> Bosnio	<input type="checkbox"/> Hawaiano	<input type="checkbox"/> Malayalam	<input type="checkbox"/> Sundanés
<input type="checkbox"/> Búlgaro	<input type="checkbox"/> Hebreo	<input type="checkbox"/> Maltés	<input type="checkbox"/> Swahili
<input type="checkbox"/> Catalán	<input type="checkbox"/> Indio	<input type="checkbox"/> Maori	<input type="checkbox"/> Sueco
<input type="checkbox"/> Cebuano	<input type="checkbox"/> Hmong	<input type="checkbox"/> Marathi	<input type="checkbox"/> Tajik
<input type="checkbox"/> ChichewaName	<input type="checkbox"/> Húngaro	<input type="checkbox"/> Mongol	<input type="checkbox"/> Tamil
<input type="checkbox"/> Chino	<input type="checkbox"/> Islandés	<input type="checkbox"/> Myanmar	<input type="checkbox"/> Telugu
(Simplificado)	<input type="checkbox"/> Igbo	(Birmano)	<input type="checkbox"/> Thai
<input type="checkbox"/> Chino	<input type="checkbox"/> Indonesio	<input type="checkbox"/> Nepalí	<input type="checkbox"/> Turco
(Tradicional)	<input type="checkbox"/> Irlandés	<input type="checkbox"/> Noruego	<input type="checkbox"/> Ukranianés
<input type="checkbox"/> Corso	<input type="checkbox"/> Italiano	<input type="checkbox"/> Pastún	<input type="checkbox"/> Urdu
<input type="checkbox"/> Croata	<input type="checkbox"/> Japonés	<input type="checkbox"/> Persa	<input type="checkbox"/> Uzbeko
<input type="checkbox"/> Checo	<input type="checkbox"/> Javanés	<input type="checkbox"/> Polaco	<input type="checkbox"/> Vietnamita
<input type="checkbox"/> Danés	<input type="checkbox"/> Kannada	<input type="checkbox"/> Portugués	<input type="checkbox"/> Galés
<input type="checkbox"/> Neerlandés	<input type="checkbox"/> Kazako	<input type="checkbox"/> Punjabí	<input type="checkbox"/> Xhosa
<input type="checkbox"/> Inglés	<input type="checkbox"/> Khmer	<input type="checkbox"/> Rumano	<input type="checkbox"/> Yiddish
<input type="checkbox"/> Esperanto	<input type="checkbox"/> Coreano	<input type="checkbox"/> Ruso	<input type="checkbox"/> Yoruba
<input type="checkbox"/> Estonio	<input type="checkbox"/> Kurdo (Kurmanji)	<input type="checkbox"/> Samoano	<input type="checkbox"/> Zulú
<input type="checkbox"/> Filipino		<input type="checkbox"/> Gaélico escocés	
<input type="checkbox"/> Finés			

També hi ha un apartat, amb el codi del widget en HTML, CSS i JavaScript que hem modificat per a canviar coses com la bandera. Altre benefici d'utilitzar este plugin és que tot aquell text que es trobe dins la classe "notranslate" no serà traduït i açò és útil per no traduir certs apartats que deuen estar igual que els han escrit els usuaris.

- **WP-Polls.** Este és el plugin que gestiona les votacions. Una vegada activat, en el panel d'administració WordPress apareixerà un apartat "Encuestas" des d'on podrem gestionar tota la configuració.



Des de “Gestionar encuestas” es podrà veure l’històric de totes les enquestes, estats i resultats. També es poden veure les estadístiques i registres.



**Gestionar encuestas**

Encuestas

ID	Pregunta	Votantes totales	Fecha/hora de inicio	Fecha/hora de finalización	Estado	Acción
No se encontraron encuestas						

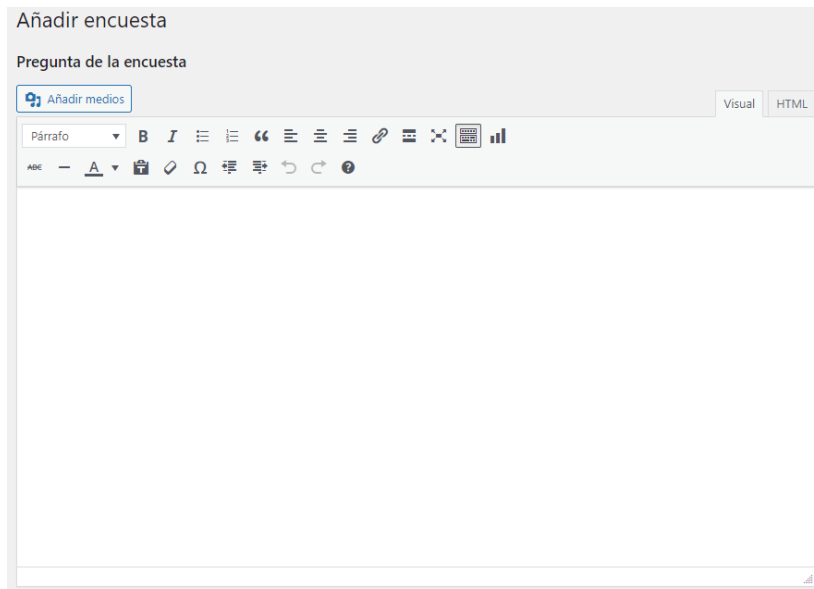
**Estadísticas de las encuestas:**

Encuestas totales:	0
Total de respuestas a las encuestas:	0
Total de votos registrados:	0
Total de votantes:	0

**Registros de las encuestas**

No hay registros de encuesta disponibles.

L’apartat “Añadir encuesta” és al que seran redirigits els administradors i membres de la junta que cliquen el botó per afegir enquestes des de la pàgina corresponent. Allí trobaran un camp on podran afegir text formatejat i imatges com a pregunta.



**Añadir encuesta**

Pregunta de la encuesta

[Añadir medios](#) Visual HTML

Párrafo

Rich text editor toolbar with icons for bold, italic, underline, link, unlink, list, indent, and other text formatting options.

ABC

Large empty text area for entering the survey question.

Més avall estan els apartats per afegir tantes respostes com siga convenient, la configuració de les respostes múltiples de l'enquesta i les dates d'inici i final.

The image shows a configuration panel titled "Respuestas a la encuesta". It contains the following elements:

- Respuestas a la encuesta:** Two input fields labeled "Respuesta 1" and "Respuesta 2", each with an "Eliminar" button to its right. Below them is an "Añadir respuesta" button.
- Respuestas múltiples a la encuesta:**
  - ¿Permitir a los usuario seleccionar más de una respuesta? with a "No" button.
  - ¿Número máximo permitido de respuestas seleccionadas? with an input field containing "1".
- Fecha de inicio/fin de la encuesta:**
  - Fecha/hora de inicio: A date and time picker showing "25", "junio", "2023", "@", "16", ":", "41", ":", "46".
  - Fecha/hora de finalización: A checkbox labeled "Que NO caduque esta encuesta".
- At the bottom, there are two buttons: "Añadir encuesta" (highlighted in blue) and "Cancelar".

En l'apartat "Opciones de la encuesta" tenim la configuració de l'estil de la barra d'enquestes.

The image shows a configuration panel titled "Opciones de la encuesta". It contains the following elements:

- Estilo de la barra de encuesta:** Three radio button options:
  - Estilo de la barra de encuesta: (default\_gradient) with a gradient bar.
  - Estilo de la barra de encuesta: (default) with a solid blue bar.
  - Usar estilo CSS: (selected) with a blue dot.
- Fondo de la barra de la encuesta:** A color picker with a hex code input field containing "#d8e1eb" and a color swatch.
- Borde de la barra de la encuesta:** A color picker with a hex code input field containing "#c8c8c8" and a color swatch.
- Alto de la barra de la encuesta:** An input field containing "8" followed by "px".
- La barra de tu encuesta se verá así:** A preview of a solid blue horizontal bar.

La configuració de l'ordre de les respostes i resultats.

**Orden de las respuestas a la encuesta**

Ordenar respuestas a la encuesta por:

Orden de ordenación de las respuestas a la encuesta:

**Orden de los resultados de la encuesta**

Ordenar resultados de encuesta por:

Orden de ordenación de los resultados de la encuesta:

I altres configuracions com a qui se li permet votar o com es mostren les enquestes actives.

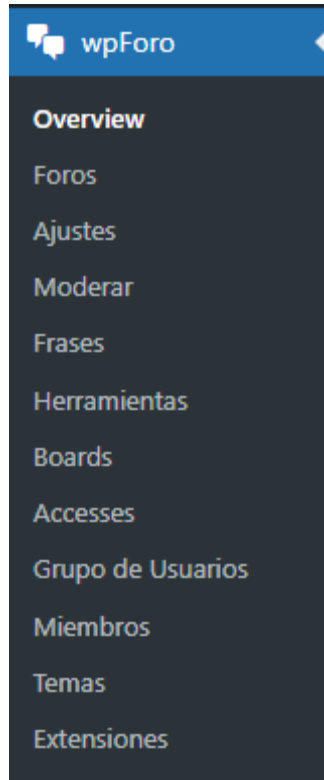
En “plantillas de encuestas” apareixen les variables per defecte que es poden utilitzar en les plantilles. També apareixen totes les plantilles, amb codi HTML i possibilitat de ser modificades, com el cos del formulari de vot, el peu de pàgina, els resultats i altres.

Com este plugin complia algun dels requisits però no tots, se li han afegit unes modificacions.

Per començar, se li ha afegit el camp “wp\_editor”, on s’afegeix la pregunta, que és el que habilita per escriure el text amb cert tipus de format i carregar arxius, ja que per defecte sols deixa escriure text pla. Tampoc mostrava les enquestes com els veïns necessitaven, per eixa raó, s’ha creat un shortcode on es mostren les enquestes en una taula i en funció del seu estat, mostren el que és convenient. Les pendents són aquelles on l’usuari encara no ha votat, per tant, a l’usuari li apareixerà i podrà clicar el botó “Votar” i es dirigirà a la pàgina de l’enquesta. Una vegada vote, li desapareixerà l’enquesta i li apareixerà un missatge on se li indicarà que ja ha votat. Quan torne a la pàgina principal de les votacions, li apareixerà l’enquesta com a “oberta” perquè vol dir que l’enquesta està activa però ja ha votat i ha d’esperar a que acabe el plaç. Una vegada algú tanque l’enquesta o es tanque sola, l’estat serà “tancada” i es podran vore els resultats. Este shortcode es cridarà des d’una pàgina creada en el panel d’administració de WordPress.



- **wpForo.** Este és un dels millors plugins gratuïts per desplegar un foro en una pàgina web WordPress. Este plugin ofereix una muntó de possibilitats. Quan s'activa ens apareixen totes les opcions en el panel d'administració de WordPress:

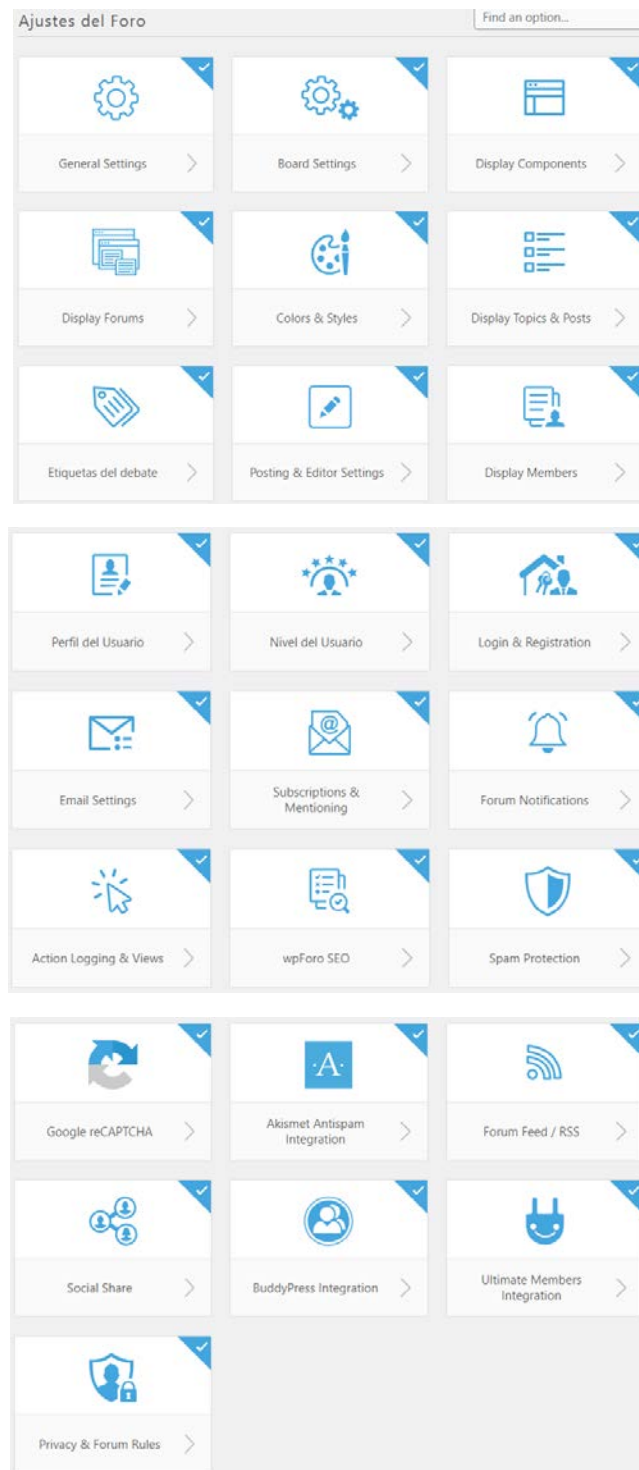


En “Overview” tenim opcions generals com informació o manteniment del foro.

En l'apartat “Foros” és on es creen les categories i els foros que tindrà l'aplicació.

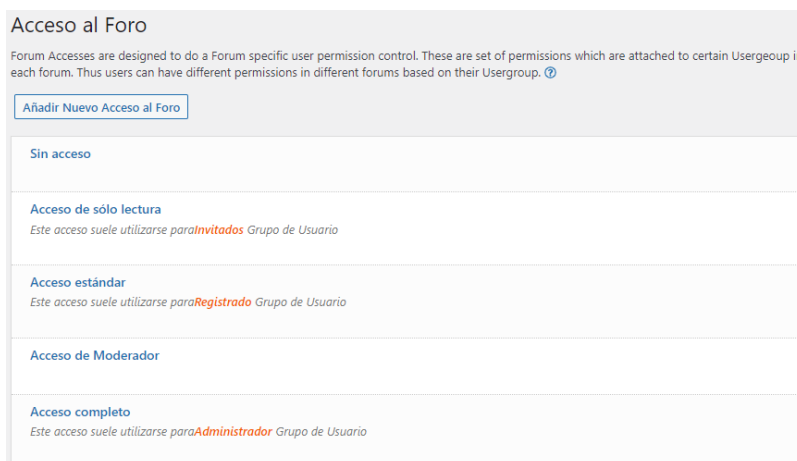


En l'apartat "Ajustes" és on ens adonem de la magnitud d'este plugin, ja que veiem la quantitat de possibilitats a configurar que ens ofereix:



En “Herramientas” podem veure les dades dels usuaris i les taules de la base de dades.

En “Accesos” es poden veure els diferents tipus d'accessos (lectura, estàndard, moderació i complet) i els usuaris que el tenen assignat.



En “Grupos de usuarios” és on podem crear grups d’usuaris segons el rol que tinguen i assignar-los als tipus d’accessos que hem vist anteriorment

En “Miembros” es pot veure informació relacionada amb el foro sobre tots els usuaris de la pàgina web i editar la seua configuració.

En “Temas” es pot canviar el tema del foro, encara que no hi ha moltes opcions.

- **classic-editor.** Este plugin instaura l’editor clàssic per a quan afegim o editem els “Custom Post Types” com són les notícies, incidències, assemblees i els anunciant.

Per altra banda està el plugin que s’ha creat i desenvolupat per cobrir certes funcionalitats de l’aplicació:

- **Baradello.** El plugin baradello, s’ha desenvolupat per crear els “Custom Post Types” requerits com són “issue”, “assembly” i “advertiser” ja que per a les notícies s’ha utilitzat el “post”, que es crea per defecte quan es llança una aplicació WordPress. També s’ha utilitzat per afegir els tipus de rols “maintenance”, “board” i “restaurant” que després s’assignen als usuaris quan es creen. A banda, al mateix temps que es crea tot el que s’ha mencionat anteriorment, es defineix la configuració i els permisos que tindran. A més, el plugin està preparat per abastir altres configuracions en un futur.

Creació del “custom post type issue”.

```
//register the issues custom post type
$labels = array(
    'name' => __('Incidencias', 'baradello-plugin'),
    'singular_name' => __('Incidencia', 'baradello-plugin'),
    'add_new' => __('Añadir nuevo', 'baradello-plugin'),
    'add_new_item' => __('Añadir nuevo issue', 'baradello-plugin'),
    'new_item' => __('Nuevo issue', 'baradello-plugin'),
    'all_items' => __('Todos los issues', 'baradello-plugin'),
    'view_item' => __('Ver issues', 'baradello-plugin'),
    'search_items' => __('Buscar issues', 'baradello-plugin'),
    'not_found' => __('No se encuentran issues', 'baradello-plugin')
);

$args = array(
    'labels' => $labels,
    'public' => true,
    'publicly_queryable' => true,
    'show_ui' => true,
    'show_in_menu' => true,
    'query_var' => true,
    'rewrite' => true,
    'capability_type' => 'post',
    'has_archive' => true,
    'hierarchical' => false,
    'menu_position' => null,
    'supports' => array( 'title', 'editor', 'thumbnail', 'excerpt', 'custom-fields', 'comments' )
);

register_post_type( 'issue', $args );
```

Creació del “custom post type advertiser”.

```

//register the advertisers custom post type
$labels2 = array(
    'name' => __('Anunciantes', 'baradello-plugin'),
    'singular_name' => __('Anunciante', 'baradello-plugin'),
    'add_new' => __('Añadir nuevo', 'baradello-plugin'),
    'add_new_item' => __('Añadir nuevo anunciante', 'baradello-plugin'),
    'new_item' => __('Nuevo anunciante', 'baradello-plugin'),
    'all_items' => __('Todos los anunciantes', 'baradello-plugin'),
    'view_item' => __('Ver anunciantes', 'baradello-plugin'),
    'search_items' => __('Buscar anunciantes', 'baradello-plugin'),
    'not_found' => __('No se encuentran anunciantes', 'baradello-plugin')
);

$args2 = array(
    'labels' => $labels2,
    'public' => true,
    'publicly_queryable' => true,
    'show_ui' => true,
    'show_in_menu' => true,
    'query_var' => true,
    'rewrite' => true,
    'capability_type' => 'post',
    'has_archive' => true,
    'hierarchical' => false,
    'menu_position' => null,
    'supports' => array( 'title', 'editor', 'thumbnail', 'excerpt', 'custom-fields' )
);

register_post_type( 'advertiser', $args2 );

```

Creació del “custom post type assembly”.

```

//register the assemblies custom post type
$labels3 = array(
    'name' => __('Asambleas', 'baradello-plugin'),
    'singular_name' => __('Asamblea', 'baradello-plugin'),
    'add_new' => __('Añadir nueva', 'baradello-plugin'),
    'add_new_item' => __('Añadir nueva assembly', 'baradello-plugin'),
    'new_item' => __('Nueva assembly', 'baradello-plugin'),
    'all_items' => __('Todas las assemblies', 'baradello-plugin'),
    'view_item' => __('Ver assemblies', 'baradello-plugin'),
    'search_items' => __('Buscar assemblies', 'baradello-plugin'),
    'not_found' => __('No se encuentran assemblies', 'baradello-plugin')
);

$args3 = array(
    'labels' => $labels3,
    'public' => true,
    'publicly_queryable' => true,
    'show_ui' => true,
    'show_in_menu' => true,
    'query_var' => true,
    'rewrite' => true,
    'capability_type' => 'post',
    'has_archive' => true,
    'hierarchical' => false,
    'menu_position' => null,
    'supports' => array( 'title', 'editor', 'thumbnail', 'excerpt', 'custom-fields' ),
    'exclude_from_search' => true
);

register_post_type( 'assembly', $args3 );

```

Rols de manteniment, junta i restaurant afegits:

```
// adding a maintenance role
add_role('maintenance',
    'Mantenimiento',
    array(
        'read' => true, // Allows a user to read
    )
);

//remove_role('board');
// adding a board role
add_role('board',
    'Miembro Junta',
    array(
        'read' => true, // Allows a user to read
        'edit_plugins' => true,
        'activate_plugins' => true,
        'publish_posts' => true,
        'edit_posts' => true,
        'edit_others_posts' => true,
        'edit_published_posts' => true,
        'delete_posts' => true,
        'delete_others_posts' => true,
        'delete_published_posts' => true,
        'manage_network_plugins' => true,
        'edit_dashboard' => true,
        'manage_polls' => true,
        'upload_files' => true,
    )
);

//remove_role('restaurant');
// adding a restaurant role
add_role('restaurant',
    'Restaurante',
    array(
        'read' => true, // Allows a user to read
        'upload_files' => true,
    )
);
```

### 3.4.2 Backend (MySQL)

El backend, en una aplicació web, és la part que l'usuari mai pot veure, és l'interior de l'aplicació que conté el servei amb la base de dades. El servei es desplega en un contenidor docker amb un fitxer docker-compose.yml

#### 3.4.2.1 Desplegament

En l'arxiu docker-compose.yml es defineixen els contenidors com a serveis. En este cas tenim el servei db, per a la base de dades, on es defineixen els següents paràmetres:

- **container\_name.** Un nom per al contenidor fàcil d'identificar per si necessitem consultar qualsevol cosa.
- **image.** Ací es defineix la imatge i versió que el contenidor descarregarà dels repositoris de Docker Hub. En este cas s'està utilitzant la versió 5.7 de MySQL.
- **volumes.** Este paràmetre és important, ja que, quan llancem el contenidor, estem mapejant dos directoris locals, molt importants per al funcionament, al contenidor mysql. El directori /docker-entrypoint-initdb.d executarà tots els scripts que continga quan es cree i llance el contenidor. És molt útil per a desplegar una base de dades amb informació i que no estiga buida al llançament. Es mapeja el directori /production/db/ini que tindrà l'script d'inici de la base de dades. També es mapeja el directori del contenidor /var/lib/mysql que conté tots els fitxers on es guarden les dades de mysql a /production/db/data per tindre la informació localment.
- **environment.** Es defineixen tres variables d'entorn molt importants com són la base de dades, l'usuari i la contrasenya que estan definides com a variables d'entorn de Linux en un fitxer .env per a que no siguen visibles a l'arxiu YML.
- **depends\_on.** Este paràmetre indica que el contenidor depen del servei db per a alçar-se. Així que fins que no s'alce db, wordpress no s'alçarà.
- **restart.** Indica que sempre que caiga es tornarà a alçar el contenidor.

```
db:
  container_name: baradello_db
  image: mysql:5.7
  restart: always
  volumes:
    - ./production/db/ini:/docker-entrypoint-initdb.d
    - ./production/db/data:/var/lib/mysql

  environment:
    MYSQL_DATABASE: ${DB_NAME}
    MYSQL_USER: ${DB_USER}
    MYSQL_PASSWORD: ${DB_PASSWORD}
    MYSQL_RANDOM_ROOT_PASSWORD: '1'
```

### 3.4.2.2 Introducció a MySQL en WordPress

El contenidor de MySQL inclou totes les dades i la lògica de l'aplicació. Quan s'instal·la WordPress per primera vegada, es creen una sèries de taules relacionals per primera vegada que formen l'estructura de la capa del Model.

A continuació s'enumeraran estes taules i es farà una breu explicació de cadascuna d'elles:

\*És important explicar que hi ha taules com `wp_users` i `wp_usermeta` on la primera taula conté les dades essencials de l'usuari i la segona conté metadades de la primera, és a dir, dades que descriuen altres dades, en este cas, dels usuaris\*

- **wp\_options.** Esta és l'única taula aïllada de la resta, és a dir, no està relacionada amb cap altra. Almacena configuració general de WordPress i dels plugins.
- **wp\_users.** Esta taula conté informació dels usuaris registrats a WordPress com el nom d'usuari, correu i contrasenya (xifrada). Quan instal·lem WordPress, hem de crear un usuari administrador amb el qual gestionarem la configuració i podrem crear més usuaris.
- **wp\_usermeta.** Esta taula conté altres dades dels usuaris com el nom complet, configuracions del plugins a nivell d'usuari o el rol assignat que descriu als usuaris emmagatzemats en `wp_users`.



- **wp\_comments.** Esta taula conté els comentaris de les pàgines i posts. Si els comentaris no estan activats en la Web, esta taula romandrà buida.
- **wp\_commentmeta.** Esta taula emmagatzema les metadades dels comentaris.
- **wp\_links.** Esta taula conté informació dels enllaços agregants a la Web però està obsoleta ja que s'utilitzava en versions antigues i es conserva per compatibilitat.
- **wp\_posts.** Esta taula conté totes les entrades del blog, les pàgines, menús i altre contingut creat per l'usuari. Inclou informació com les dates o l'autor.
- **wp\_postmeta.** Esta taula té les metadades de les entrades i les pàgines com les etiquetes o imatges destacades.
- **wp\_terms.** Esta taula emmagatzema categories i etiquetes de les publicacions, pàgines i enllaços.
- **wp\_term\_taxonomy.** Esta taula conté més dades de la taula wp\_terms.
- **wp\_term\_relationship.** Esta taula relaciona les etiquetes amb les publicacions, pàgines i enllaços.

### 3.4.2.3 Taules

Per accedir a la consola bash [13] d'un contenidor, ho podem fer amb la instrucció:

**docker exec -it <nom\_contenidor> /bin/bash**

```
joanma@vmi1026407:~$ docker exec -it baradello_db /bin/bash
bash-4.2#
```

Si al contenidor tenim desplegada una base de dades mysql i volem accedir a la seua consola ho podem fer amb la següent instrucció:

**mysql -user=<nom\_usuari> -password=<contrasenya>**

```
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 98308
Server version: 5.7.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Per a seleccionar la base de dades s'ha de fer amb la instrucció:

**use <nom\_base\_dades>**

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Amb la instrucció **show tables**; podem llistar totes les taules:

```
mysql> show tables;
+-----+
| Tables_in_baradellogelat |
+-----+
wp_actionscheduler_actions
wp_actionscheduler_claims
wp_actionscheduler_groups
wp_actionscheduler_logs
wp_commentmeta
wp_comments
wp_links
wp_options
wp_pollsa
wp_pollsip
wp_pollsq
wp_postmeta
wp_posts
wp_term_relationships
wp_term_taxonomy
wp_termmeta
wp_terms
wp_trp_dictionary_es_es_val
wp_trp_gettext_es_es
wp_trp_gettext_val
wp_trp_original_meta
wp_trp_original_strings
wp_usermeta
wp_users
wp_wpforo_accesses
wp_wpforo_activity
wp_wpforo_boards
wp_wpforo_bookmarks
wp_wpforo_follows
wp_wpforo_forums
wp_wpforo_languages
wp_wpforo_logs
wp_wpforo_phrases
wp_wpforo_post_revisions
wp_wpforo_postmeta
wp_wpforo_posts
wp_wpforo_profiles
wp_wpforo_reactions
wp_wpforo_subscribes
wp_wpforo_tags
wp_wpforo_topics
wp_wpforo_usergroups
wp_wpforo_visits
wp_wpmailsmtplib_debug_events
wp_wpmailsmtplib_tasks_meta
```

Les taules wp\_trp\_\* són les del plugin **GTranslate**.

Les taules wp\_wpforo són del plugin del **wpForo**.

Les taules del wp\_wpmailsmtplib són del plugin del **SMTP Gmail**.

Les taules de wp\_poll\* són del plugin de les enquestes **WP-Polls**.

### 3.4.3 Reverse-Proxy (Caddy)

El Reverse-Proxy web és un servici que es desplega per a estar escoltant les peticions que arriben al servidor per part dels clients i redirigir-les cap a l'aplicació. És l'intermediari entre els clients i el servici de la pàgina web. Serveix per a balancejar les càrregues de les peticions o per afegir seguretat a l'aplicació i gestionar les certificats SSL, entre altres coses.

#### 3.4.3.1 Desplegament

En l'arxiu `docker-compose.yml` definim el servici/contenedor `caddy` per al Reverse-Proxy amb els següents paràmetres:

- **container\_name.** Definim el nom "caddy" per identificar millor el contenidor quan ho requerim.
- **imatge.** La imatge del repositori de Docker d'on es formarà el contenidor i l'última versió que hi haja.
- **ports.** Es mapegen els ports 80 i 443 del contenidor als ports 80 i 443 del servidor. És necessari mapejar estos ports ja que són els dels serveis HTTP i HTTPS que estan escoltant en qualsevol petició web al domini `baradellogelat.com` o `www.baradellogelat.com`. Després en la configuració ja es redirigiran les peticions al contenidor WordPress.
- **volumes.** Es mapegen tres volums i un fitxer. El directoris de configuració, dades i certificats i l'arxiu `Caddyfile` es mapejaran al directori `/production/caddy` del servidor per a modificar configuracions.
- **restart.** `Unless-stopped` vol dir que si el contenidor es para manualment també s'haurà d'alçar manualment.
- **depends\_on.** Depèn dels contenidors de `wordpress` i la base de dades, ja que no té sentit que estiga en marxa si els altres no funcionen.

```
caddy:
  image: caddy:latest
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - ./production/caddy/Caddyfile:/etc/caddy/Caddyfile:ro
    - ./production/caddy/certs:/etc/caddy/certs
    - ./production/caddy/data:/data
    - ./production/caddy/config:/config
  restart: unless-stopped
  container_name: caddy
  depends_on:
    - wordpress
    - db
```

### 3.4.3.2 Configuració

Una de les raons per les quals s'ha escollit Caddy és la senzillesa de la sintaxis en la seua configuració. L'arxiu que s'utilitza es diu Caddyfile i ha d'estar en el directori `/etc/caddy` del servidor. Per eixe motiu, es mapeja el fitxer Caddyfile al directori esmentat. També, altre punt important, és que Caddy té capacitat per gestionar els certificats TLS i desplegar la web en HTTPS. En el directori `/etc/caddy/certs` del servidor tindrem el certificat i la clau privada amb l'extensió "pem".

Els directoris de `/data` i `/config` es generen automàticament i guarden fitxers de dades i configuració. Es mapegen per tindre'ls controlats i guardats però no haurem de fer cap modificació dins d'ells.

El fitxer és el següent:

```
https://baradellogelat.com {
    tls /etc/caddy/certs/cert.pem /etc/caddy/certs/privkey.pem
    reverse_proxy wordpress:
}

https://www.baradellogelat.com {
    tls /etc/caddy/certs/cert.pem /etc/caddy/certs/privkey.pem
    reverse_proxy wordpress:
}
```

En el fitxer es defineix que tot el que arribe als dominis <https://baradellogelat.com> i <https://www.baradellogelat.com> farà un reverse\_proxy al contenidor wordpress al qual no és necessari especificar-li el port ja que la comunicació serà per la xarxa interna de Docker. A més, també s'especifica el certificat amb la instrucció `tls <certificat> <clau_privada>`. Amb eixa configuració tan senzilla tindrem un servidor fent de reverse-proxy amb Caddy per a la nostra pàgina web.

### 3.4.4 WP-CLI

WP-CLI és la interfície de línia de comandaments per a WordPress que permet administrar la pàgina web sense haver d'accedir a través del panel web. En el nostre cas, este contenidor sols serà utilitzat a l'hora del desplegament on executarà un script bash per a crear tots els usuaris.

#### 3.4.4.1 Desplegament

El desplegament d'este contenidor serà igual que els anteriors, definint un servei al fitxer docker-compose.yml.

- **image.** Ací es defineix la imatge i versió que el contenidor descarregarà dels repositoris de Docker Hub. En este cas s'està utilitzant la versió cli de wordpress.
- **volumes.** Este paràmetre és molt important perquè els comandaments de wp-cli s'executen en el directori arrel d'un projecte WordPress. Per eixe motiu, es mapeja el directori arrel del projecte en el servidor de producció al directori arrel d'un projecte WordPress com és el /var/www/html. Sense este mapeig, no funcionen les instruccions de wp-cli.
- **environment.** Es defineixen quatre variables d'entorn molt importants com són el host, l'usuari, la contrasenya i el nom de la base de dades. El host no és important xifrar-lo, ja que és el contenidor de la base de dades, però l'usuari, la contrasenya i el nom, per seguretat, s'han definit com a variables d'entorn de Linux en un fitxer .env per a que no siguin visibles a l'arxiu YML.
- **command.** Com el seu nom indica, executa una instrucció quan es desplega el contenidor. La instrucció en bash és "sleep 60 && /var/html/user.sh".

```
wp-cli:
  image: wordpress:cli
  volumes:
    - ./production/www:/var/www/html/
  depends_on:
    - wordpress
    - db
  environment:
    WORDPRESS_DB_HOST: db
    WORDPRESS_DB_USER: ${DB_USER}
    WORDPRESS_DB_PASSWORD: ${DB_PASSWORD}
    WORDPRESS_DB_NAME: ${DB_NAME}
  command: /bin/sh -c "sleep 60 && /var/www/html/users.sh"
```

### 3.4.4.2 Funcionament

El funcionament d'este contenidor és ben senzill. Serveix per al desplegament dels serveis i ja no es torna a utilitzar. En el desplegament executa dos comandaments: El primer és un sleep de 60 segons que serveix per esperar mentre es creen els altres contenidors i donar temps a que s'execute l'script que carrega i configura la base de dades. El segon és l'execució de l'script users.sh que vorem a continuació.

Users.sh és un script que llig i recorre un arxiu csv, on estan declarats tots els usuaris amb un format prèviament definit, i que executa la següent instrucció de wp-cli per a cada usuari:

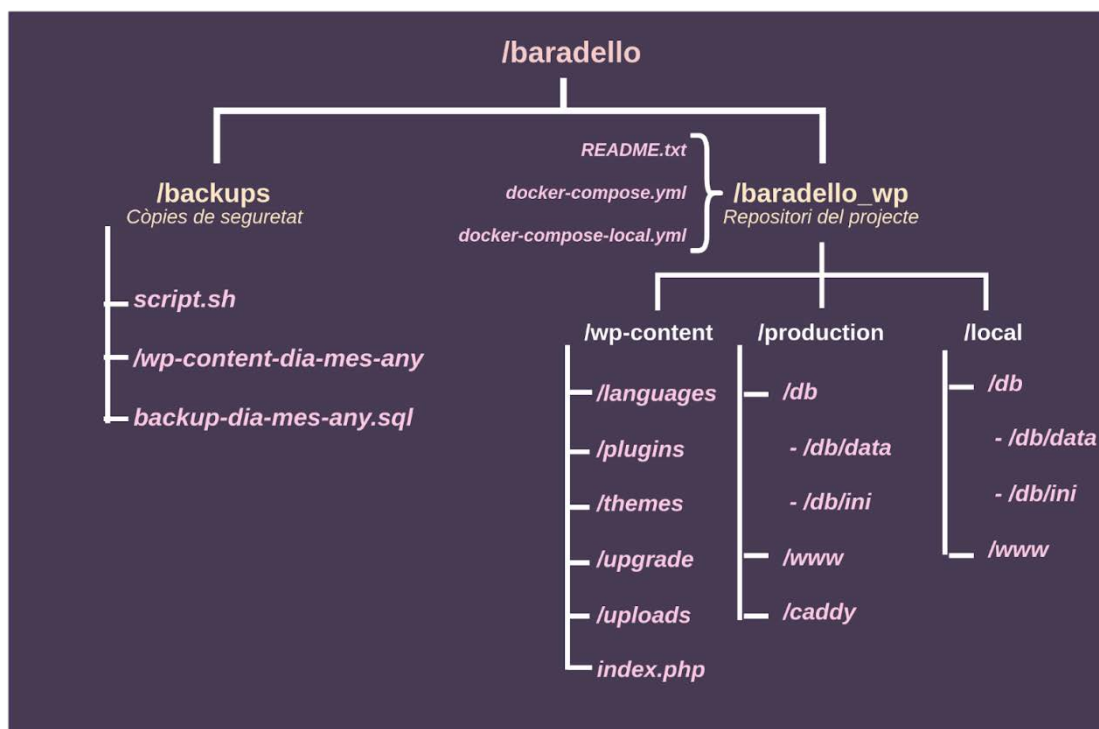
```
"wp user create <user-login> <user-mail --role=<rol> --first_name=<nom> --user_pass=<contasenya>"
```

```
#!/bin/bash
# Propósito: Leer archivos CSV separados por comas
# Autor: Joanma
# -----
INPUT=censo-usuarios.csv
IFS=','
[ ! -f $INPUT ] && { echo " $INPUT file not found" ; exit 99 ; }
while read Surname Name Street Number Street2 Number2 Email Role
do
    # Clearing the quotation marks
    Surname="${Surname%\"}"
    Surname="${Surname#\}"
    Name="${Name%\"}"
    Name="${Name#\}"
    Name="${Name//[[[:blank:]]]}/"
    Name="${Name#\}"
    Street="${Street%\"}"
    Street="${Street#\}"
    Number="${Number%\"}"
    Number="${Number#\}"
    Number="${Number//[[[:blank:]]]}/"
    Street2="${Street2%\"}"
    Street2="${Street2#\}"
    Number2="${Number2%\"}"
    Number2="${Number2#\}"
    Number2="${Number2//[[[:blank:]]]}/"
    #echo "Surname :$Surname."
    #echo "Name :$Name."
    #echo "Street :$Street. "
    #echo "Number :$Number. "
    Email2='@baradellogelat.com'
    if [ -v $Email ]; then
        wp user create $Street$Number $Street$Number$Email2 --role=$Role --first_name=$Street2$Number
        #echo "Email :$Street$Number$Email2. "
        #echo "username: $Street$Number"
        #echo "name: $Street2$Number2"
    else
        wp user create $Street$Number $Email --role=$Role --first_name=$Street2$Number2 --user_pass=$
        #echo "Email :$Email. "
        #echo "username: $Street$Number"
        #echo "name: $Street2$Number2"
    fi
    #echo "Role :$Role. "
done < $INPUT
```



### 3.5 Estructura i disseny del servidor

A continuació va a mostrar-se un esquema de l'estructura dels directoris i fitxers més importants que componen el projecte:



El directori arrel del projecte **/baradello** conté dos subdirectoris:

- **/backups**. Este directori conté les còpies de seguretat dels últims cinc dies de la base de dades, amb un script sql, i del directori **/wp-content**, on està el tot el projecte, però que també té la carpeta **/uploads** amb tots els fitxers carregats pels usuaris.
- **/baradello\_wp**. Este directori és un repositori git que conté el projecte al complet: el WordPress, la base de dades, el caddy i els docker-compose de producció i local.

Una part del codi s'ha dividit en dos parts: producció (aplicació en funcionament) i local (desenvolupament en local). La part comú del codi és el directori **wp-content** on està el tema, els plugins i els fitxers que carreguen els usuaris que es pot utilitzar tant producció com en local ja que no té fitxers de configuració que puguin afectar o marcar la diferència del servidor a un ordinador local.

La resta del projecte s'ha hagut de dividir en dos parts perquè un projecte en producció té les seues particularitats respecte al procés mentre està en

desenvolupament en un ordinador local. Per començar, en la carpeta production és menester la configuració de caddy, a diferència del projecte en local, així que sols estarà en producció. El directori /www és l'arrel d'un projecte WordPress i s'ha diferenciat perquè en producció conté els fitxers users.sh i censo-usuarios.csv que són imprescindibles per a la creació dels usuaris en producció quan s'alcen els contenidors, cosa innecessària en local. També s'ha de remarcar que l'arxiu wp-config.php ha de ser diferent en producció ja que conté informació sobre la base de dades, que també és diferent i, sobre el domini i el HTTPS, configuració que no s'utilitza en local ja que el domini és localhost i no és necessària la configuració d'HTTPS. Altra diferència són les bases de dades. El fitxer sql que carrega la base de dades quan s'inicia el contenidor és diferent en producció del que inicia la b.b.d.d. local perquè té paràmetres com el domini o altres configuracions pròpies de l'entorn en el que s'està utilitzant. També les dos bases de dades utilitzen credencials diferents.

Han sigut moltes les diferències, entre el desplegament d'un projecte en local i un projecte en producció, que ens han portat a separar el repositori d'esta manera, intentant conservar aquelles carpetes que són comuns i diferenciant-les de les parts que havien d'anar separades

## 3.6 Backups

Les còpies de seguretat són una part fonamental de qualsevol sistema ja que permeten recuperar i restaurar un sistema que haja pogut patir un problema de qualsevol tipus. Per a una pàgina web d'estes característiques s'ha cregut suficient realitzar una còpia una vegada al dia. En este projecte, s'ha estudiat quines serien les parts necessàries a l'hora de recuperar o de desplegar el projecte WordPress en qualsevol punt. La primera és molt comuna: la base de dades. La segona és més comuna tindre-la en còpia de seguretat en un repositori, però no és tan comú fer una còpia cada dia: El directori wp-content, que conté tot el codi programat al tema i als plugins. El motiu de realitzar una còpia al dia d'este directori és que també conté la carpeta uploads on es guarden tots els fitxers carregats pels usuaris, normalment imatges i arxius de text.

Per a realitzar una còpia en un moment determinat, s'ha fet ús de les ferramentes cron i crontab natives de Linux. Cron és un dimoni, és a dir, un procés que s'executa en segon pla, que es posa en funcionament quan s'alça el servidor i comprova si existeix alguna tasca per executar. Crontab és el fitxer, amb un determinat format, que llig cron per executar la tasca corresponent.

Amb la instrucció "crontab -e" accedim al fitxer i definim quan volem que s'execute la tasca. En este cas s'ha definit que l'usuari root executarà tots els dies a les 02:00h el fitxer /baradello/backups/script.sh

```
0 2 * * * root /baradello/backups/script.sh
```

L'script en bash és el següent:

```
#!/bin/bash
DATE=$(date +%d-%m-%Y)
BACKUP_DIR="/baradello/backups"
docker exec baradello_db /usr/bin/mysqldump -u ${DB_USER} --password=${DB_PASSWORD} --no-tablespaces ${DB_NAME} > $BACKUP_DIR/backup-$DATE.sql
cp -r /baradello/baradello_wp/wp-content /baradello/backups/wp-content-$DATE
find $BACKUP_DIR/backup* -mtime +5 -exec rm {} \;
find $BACKUP_DIR/wp-content* -mtime +5 -exec rm -rf {} \;
```

**DATE=\$(date +%d-%m-%Y)**: Obtenim la data actual.

**BACKUP\_DIR="/baradello/backups"**: Definim el directori on es guardarà,

**docker exec baradello\_db /usr/bin/mysqldump -u \${DB\_USER} --password=\${DB\_PASSWORD} --no-tablespaces \${DB\_NAME} > \$BACKUP\_DIR/backup-\$DATE.sql**: Fem la còpia del fitxer sql i el guardem.

**cp -r /baradello/baradello\_wp/wp-content /baradello/backups/wp-content-\$DATE**: Fem la còpia del directori wp-content a la carpeta dels backups.

**find \$BACKUP\_DIR/backup\* -mtime +5 -exec rm {} \;** Comprovem si existeix alguna còpia de la base de dades més antiga de cinc dies i, si és el cas, l'eliminem.

**find \$BACKUP\_DIR/wp-content\* -mtime +5 -exec rm -rf {} \;** Comprovem si existeix alguna còpia del directori wp-content més antiga de cinc dies i, si és el cas, l'eliminem.

## 3.7 Seguretat

Implementar certes pautes de seguretat és una part fonamental de qualsevol sistema informàtic i més quan contenen informació sensible dels usuaris. Per eixe motiu, s'ha escanejat la nostra pàgina web, una vegada estava desplegada, per trobar vulnerabilitats.

S'ha utilitzat un escàner general de codi obert anomenat nuclei [14]. El resultat ha sigut el següent:

```
joanna@joannapc:~/Desktop/tfg$ nuclei -target "https://www.baradellogelat.com"

nuclei v2.9.0
projectdiscovery.io

[INF] nuclei-templates are not installed, installing...
[INF] Successfully downloaded nuclei-templates (v9.4.1) to /home/joanna/nuclei-templates. GoodLuck!
[INF] Using Nuclei Engine 2.9.0 (latest)
[INF] Using Nuclei Templates 9.4.1 (latest)
[INF] Templates added in last update: 69
[INF] Templates loaded for scan: 5769
[INF] Targets loaded for scan: 1
[INF] Templates clustered: 1039 (Reduced 961 Requests)
[tech-detect:font-awesome] [http] [info] https://www.baradellogelat.com
[tech-detect:caddy] [http] [info] https://www.baradellogelat.com
[tech-detect:php] [http] [info] https://www.baradellogelat.com
[metatag-cms] [http] [info] https://www.baradellogelat.com [WordPress 6.1.1]
[php-detect] [http] [info] https://www.baradellogelat.com [7.4.33]
[apache-detect] [http] [info] https://www.baradellogelat.com [Caddy Apache/2.4.54 (Debian)]
[wordpress-accessible-wpconfig] [http] [medium] https://www.baradellogelat.com/.wp-config.php.swp
[INF] Using Interactsh Server: oast.pro
[wordpress-xmlrpc-listmethods] [http] [info] https://www.baradellogelat.com/xmlrpc.php
[wordpress-user-enun] [http] [info] https://www.baradellogelat.com/?author=1 [author/joagrefa]
[wordpress-detect:version_by_js] [http] [info] https://www.baradellogelat.com [6.1.1]
[ssl-dns-names] [ssl] [info] www.baradellogelat.com:443 [www.baradellogelat.com,baradellogelat.com]
[ssl-tlsuer] [ssl] [info] www.baradellogelat.com:443 [GlobalSign nv-sa]

[wordpress-xmlrpc-file] [http] [info] https://www.baradellogelat.com/xmlrpc.php
[http-missing-security-headers:access-control-allow-origin] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:access-control-max-age] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:access-control-allow-methods] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:access-control-allow-headers] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:clear-site-data] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:cross-origin-resource-policy] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:x-content-type-options] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:referrer-policy] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:access-control-allow-credentials] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:access-control-expose-headers] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:strict-transport-security] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:content-security-policy] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:permissions-policy] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:x-frame-options] [http] [info] https://www.baradellogelat.com
[http-missing-security-headers:cross-origin-opener-policy] [http] [info] https://www.baradellogelat.com
[wp-license-file] [http] [info] https://www.baradellogelat.com/license.txt
[robots-txt-endpoint] [http] [info] https://www.baradellogelat.com/robots.txt
[tls-version] [ssl] [info] www.baradellogelat.com:443 [tls12]
[tls-version] [ssl] [info] www.baradellogelat.com:443 [tls13]
[wordpress-gtranslate:outdated_version] [http] [info] https://www.baradellogelat.com/wp-content/plugins/gtranslate/readme.txt [2.9.14] [last_version="3.0.2"]
[waf-detect:apachegenerleic] [http] [info] https://www.baradellogelat.com/
[wordpress-wp-mail-sntp:outdated_version] [http] [info] https://www.baradellogelat.com/wp-content/plugins/wp-mail-sntp/readme.txt [3.6.1] [last_version="3.7.0"]
[cve-2017-5407:username] [http] [medium] https://www.baradellogelat.com/wp-json/wp/v2/users/ [joagrefa]
[openssh-detect] [network] [info] www.baradellogelat.com:22 [SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.5]
[wordpress-classic-editor:detected_version] [http] [info] https://www.baradellogelat.com/wp-content/plugins/classic-editor/readme.txt [1.6.2] [last_version="1.6.2"]
```

També s'ha utilitzat altre escàner de codi obert [15], per comprovar quines capçaleres de resposta ens falten:

```
joanma@joanmapc:~/Desktop/tfg/SCT_v1.3.0_linux_386$ ./sct_v1.3.0_linux_386 -url https://www.baradellogelat.com
Checking https://www.baradellogelat.com for security configuration issues
Tested on: Fri, 31 Mar 2023 12:45:51 CEST

== HEADER AUDIT ==
[*] Content-Security-Policy (Not present)
[*] Strict-Transport-Security (Not present)
[*] X-Frame-Options (Not present)
[*] X-Content-Type-Options (Not present)
[*] Referrer-Policy (Not present)

== RAW HEADERS ==
Alt-Svc: h3=":443"; ma=2592000
Date: Fri, 31 Mar 2023 10:45:51 GMT
Server: Caddy
Apache/2.4.54 (Debian)
Vary: Accept-Encoding
X-Powered-By: PHP/7.4.33
Content-Type: text/html; charset=UTF-8
Link: <https://www.baradellogelat.com/wp-json/>; rel="https://api.w.org/"

== COOKIE AUDIT ==
```

La primera vulnerabilitat que trobem, són les pàgines visibles, que deurien d'estar restringides.

La solució està en restringir-les des de l'arxiu .htaccess:

```
<files wp-config.php>
order allow,deny
deny from all
</files>

<files wp-config.php.swp>
order allow,deny
deny from all
</files>

<files license.txt>
order allow,deny
deny from all
</files>

<files xmlrpc.php>
order allow,deny
deny from all
</files>

<files *.txt>
order allow,deny
deny from all
</files>

<Files wp-json>
order deny,allow
deny from all
</Files>
```

L'anterior solució, fa que ens redirigisca a una pàgina per defecte 403 on es dona massa informació sobre el nostre sistema:

## Forbidden

You don't have permission to access this resource.

Apache/2.4.54 (Debian) Server at www.baradellogelat.com Port 80

Per eixa raó, crearem l'arxiu 403.php igual que el 404.php i afegirem esta instrucció al .htaccess:

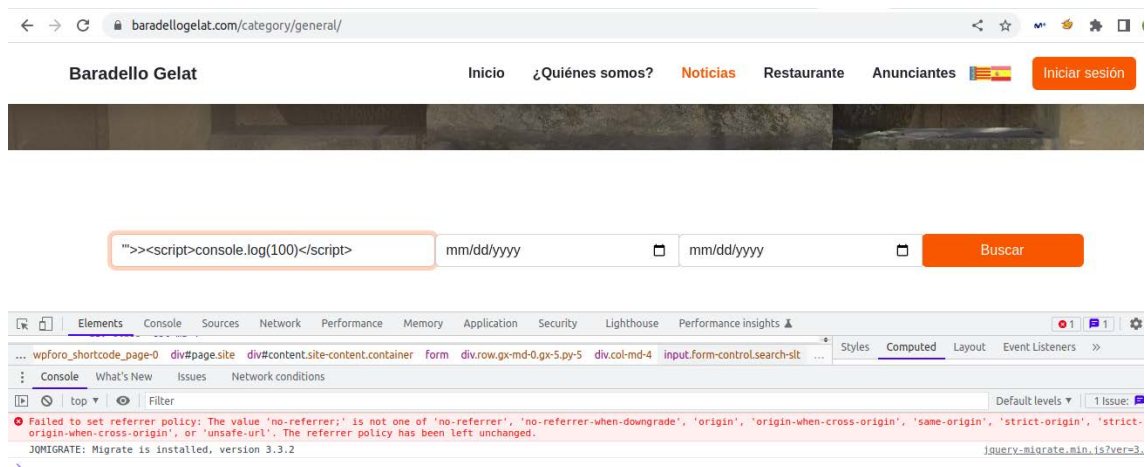
```
ErrorDocument 403 /403.php
```

Altra vulnerabilitat és que falten les cinc capçaleres imprescindibles:

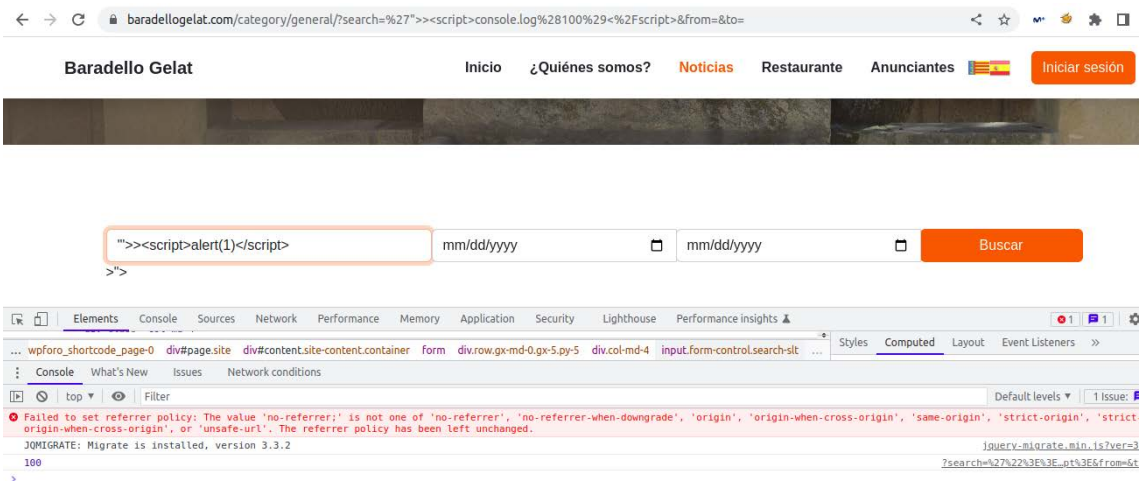
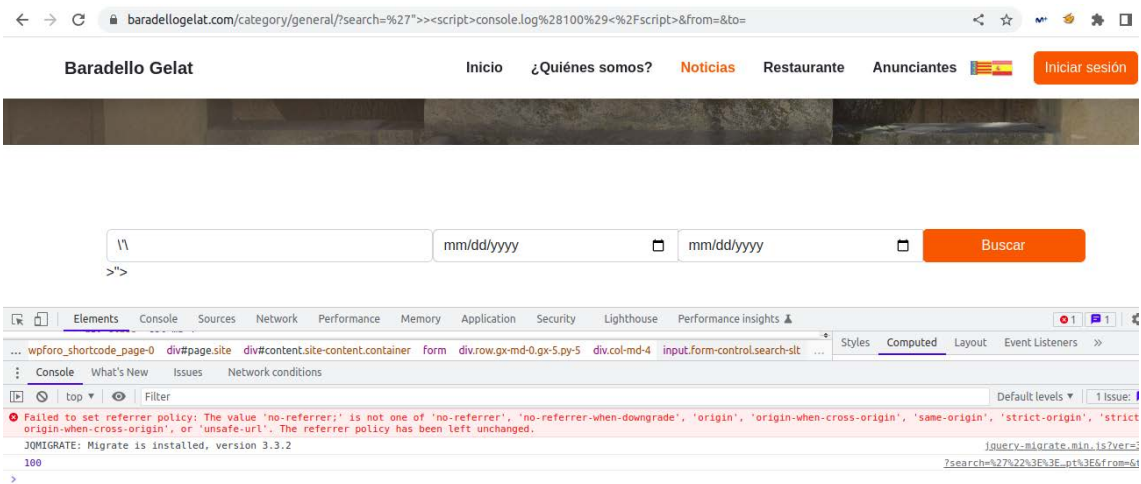
- Strict-Transport-Security
- X-Frame-Options
- X-Content-Type-Options
- Referrer-Policy
- Content-Security-Policy

Un dels problemes de no tindre la capçalera Content-Security-Policy és que podem sofrir atacs XSS (Cross-Site Scripting) [16] que són un tipus d'injecció d'scripts maliciosos, per exemple a través dels camps d'entrades de dades. En les següents captures es pot comprovar com la pàgina era susceptible d'este tipus d'atacs.

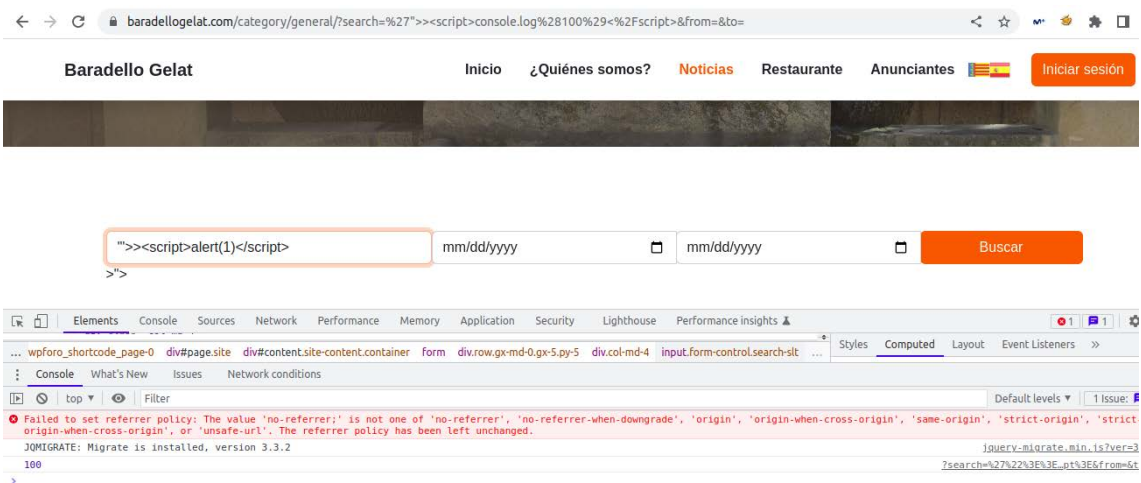
En esta captura observem com s'introdueix un script, escapant codi HTML:



En esta captura comprovem en la consola de JavaScript que l'script s'ha executat correctament:



En esta captura llancem una alerta:



En esta captura observem que també ha pres efecte la injecció:





La solució a les capçaleres passa per definir-les al Caddyfile en el Reverse-Proxy:

```
https://baradellogelat.com {
    tls /etc/caddy/certs/cert.pem /etc/caddy/certs/privkey.pem
    reverse_proxy wordpress:
    header {
        Strict-Transport-Security "max-age=31536000 ; includeSubDomains"
        X-Content-Type-Options nosniff;
        X-XSS-Protection "1; mode=block;"
        X-Robots-Tag none;
        X-Frame-Options SAMEORIGIN;
        Referrer-Policy no-referrer;
        Content-Security-Policy "img-src *;"
        -Server
        -x-powered-by
    }
}

https://www.baradellogelat.com {
    tls /etc/caddy/certs/cert.pem /etc/caddy/certs/privkey.pem
    reverse_proxy wordpress:
    header {
        Strict-Transport-Security "max-age=31536000 ; includeSubDomains"
        X-Content-Type-Options nosniff;
        X-XSS-Protection "1; mode=block;"
        X-Robots-Tag none;
        X-Frame-Options SAMEORIGIN;
        Referrer-Policy no-referrer;
        Content-Security-Policy "img-src *;"
        -Server
        -x-powered-by
    }
}
```

I reiniciar el contenidor caddy per a que s'aplique la configuració.

Tornem a escanejar la pàgina web i estan totes les capçaleres correctament:

```
joanma@joanmapc:~/Desktop/tfg/SCT_v1.3.0_linux_386$ ./sct_v1.3.0_linux_386 -url https://www.baradellogelat.com

Checking https://www.baradellogelat.com for security configuration issues
Tested on: Fri, 31 Mar 2023 12:47:17 CEST

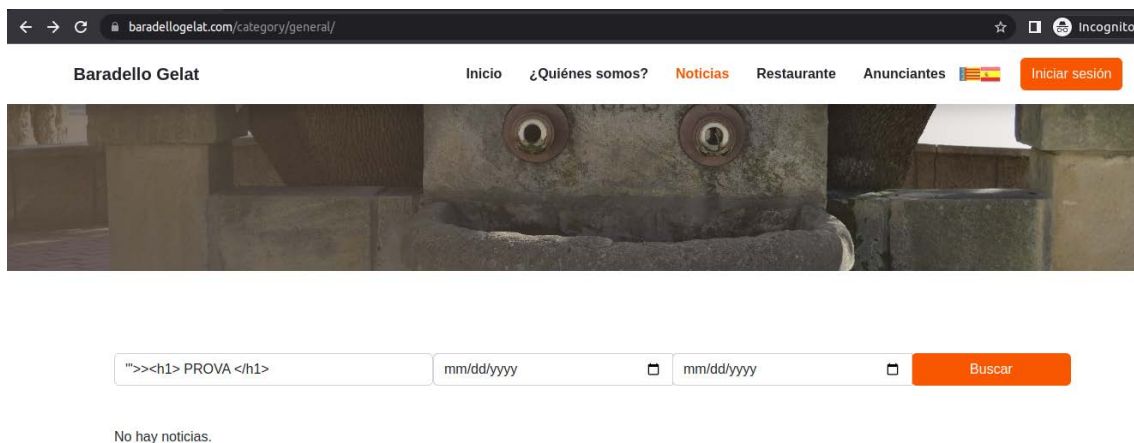
== HEADER AUDIT ==
[✓] Strict-Transport-Security [max-age=31536000 ; includeSubDomains]
[✓] X-Frame-Options [SAMEORIGIN;]
[✓] X-Content-Type-Options [nosniff;]
[✓] Referrer-Policy [no-referrer;]
[✓] Content-Security-Policy [script-src 'self'];

== RAW HEADERS ==
X-Content-Type-Options: nosniff;
Alt-Svc: h3=":443"; ma=2592000
Date: Fri, 31 Mar 2023 10:47:18 GMT
Referrer-Policy: no-referrer;
Content-Type: text/html; charset=UTF-8
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: SAMEORIGIN;
X-Robots-Tag: none;
X-Xss-Protection: 1; mode=block;
Content-Security-Policy: script-src 'self';
Link: <https://www.baradellogelat.com/wp-json/>; rel="https://api.w.org/"
Vary: Accept-Encoding

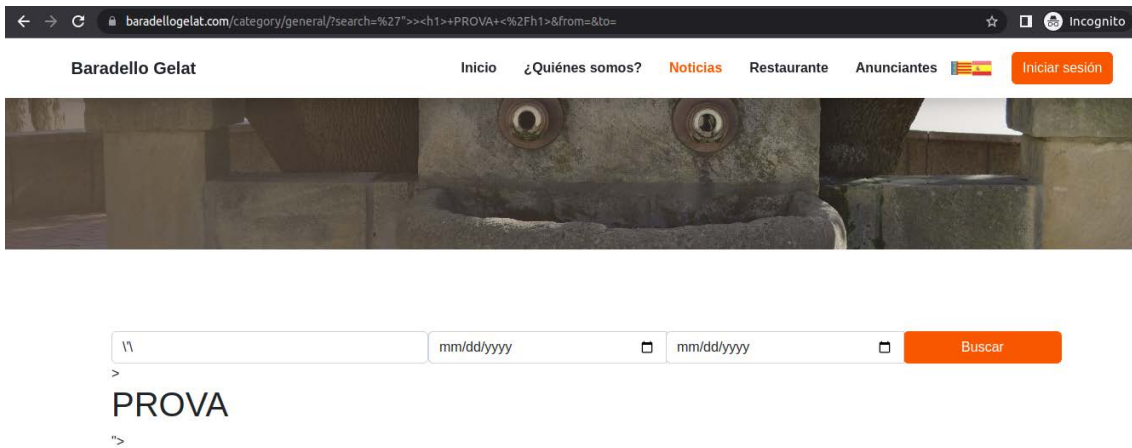
== COOKIE AUDIT ==

== RAW COOKIES ==
```

S'han trobat també vulnerabilitats d'injecció de codi HTML. En la següent captura es pot comprovar com s'ingereix de forma molt pareguda al codi JavaScript:



En esta captura veiem com apareix l'etiqueta a la web:



La solució passa per utilitzar la funció `esc_html()`, nativa de WordPress, que escapa les cadenes de text per a que no siguin interpretades com HTML. Per exemple, esta funció converteix el caràcter “<” en “&lt;”. Esta funció ha de ser cridada en els camps on hi ha entrades de dades de l’usuari.

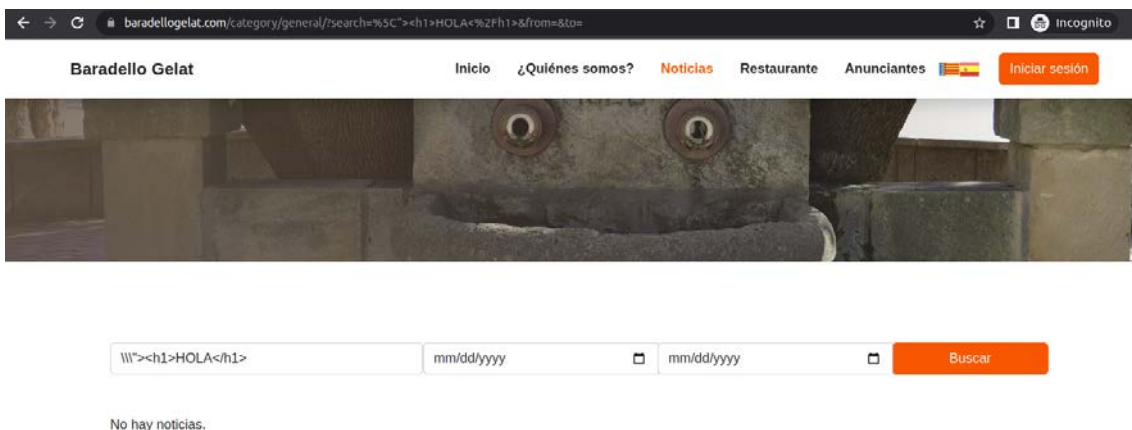
A continuació es mostren les solucions aplicades:

```
<input type="text" name="search" class="form-control search-slt" placeholder="Texto a buscar" value="<?= isset($_REQUEST['search'])?esc_html($_REQUEST['search']) : '' ?>"
```

```
<input type="text" name="search" class="form-control search-slt" placeholder="Texto a buscar" value="<?= esc_html($_REQUEST['search']) ?>"
```

```
<input type="text" name="search" class="form-control search-slt" placeholder="Texto a buscar" value="<?= esc_html($_REQUEST['search']) ?>"
```

I la comprovació de que ja s’interpreten les entrades com HTML:



## 4. Resultat i tour per l'aplicació

En este apartat van a realitzar-se captures de pantalles de com ha quedat l'aplicació per vistes.

Quan l'usuari accedeix a la pàgina web, la primera vista en la que es troba és la d'inici:



¿Dónde estamos?



### Contacto

Nombre

Email


Texto

**ENVIAR**


Si vol consultar les notícies, vorà una vista semblant a esta:

Baradello Gelat

[Início](#) [¿Quiénes somos?](#) [Notícies](#) [Restaurante](#) [Anunciantes](#) [Iniciar sesión](#)



Texto a buscar

 **Presentación**  
22/05/2023  
Buenas tardes, vecinos. Desde la Junta estamos tratando de activar el funcionamiento de esta página web, para poder trasladaros desde ella algunas noticias de interés. Os pedimos un poco de paciencia!

© 2023 - Baradello Gelat

Per a iniciar sessió, es vorà una vista com esta:

Baradello Gelat Inicio ¿Quiénes somos? Noticias Restaurante Anunciantes ES Iniciar sesión

Nombre de usuario

Contraseña

[¿te olvidado la contraseña?](#)

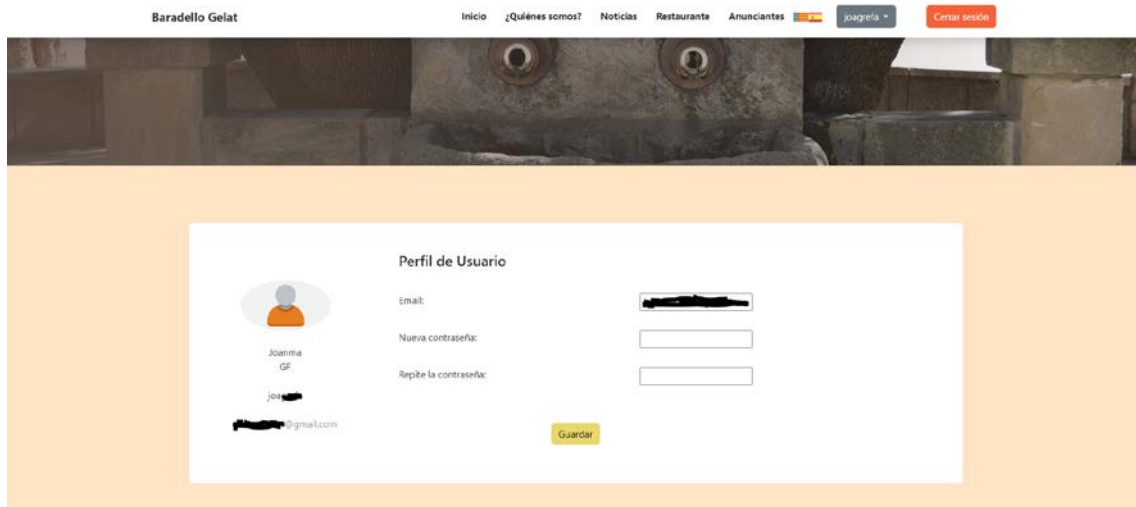
Entrar

© 2023 - Baradello Gelat

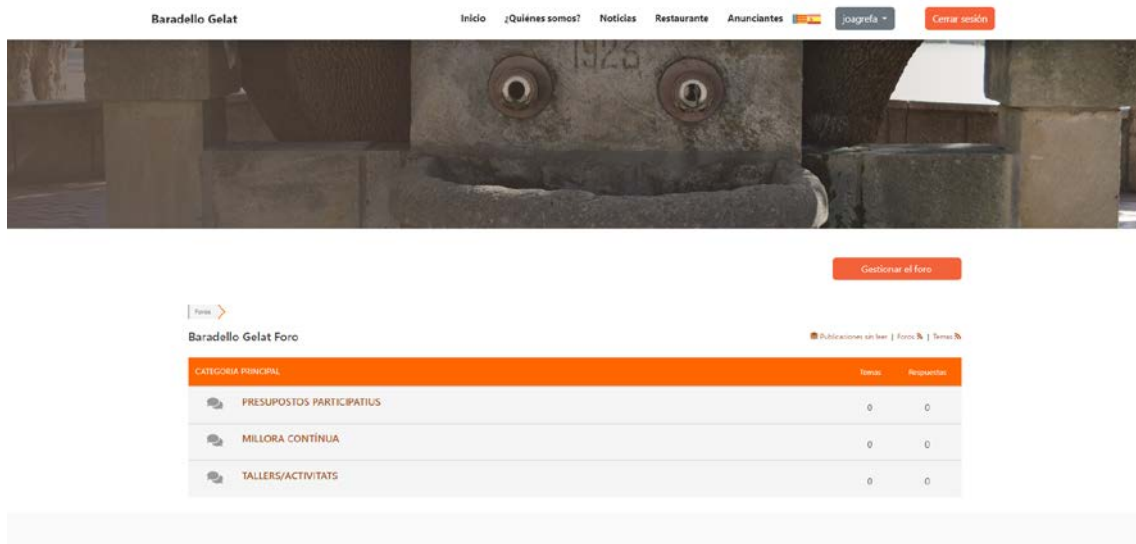
Una vegada l'usuari inicié sessió, podrà visualitzar el menú desplegable amb les opcions de la intranet.



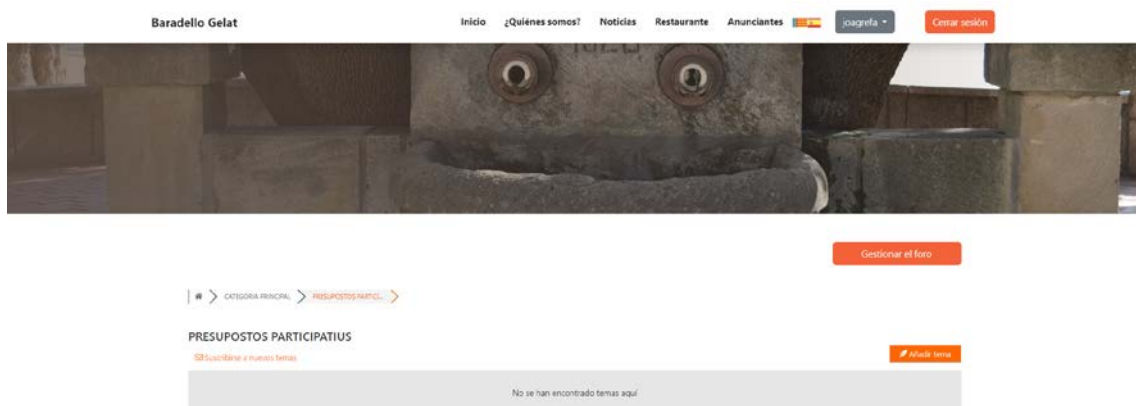
Podrà visitar el seu perfil, on podrà canviar l'email i la contrasenya.



Podrà anar a la pàgina del foro on podrà accedir als subforos que hi ha.

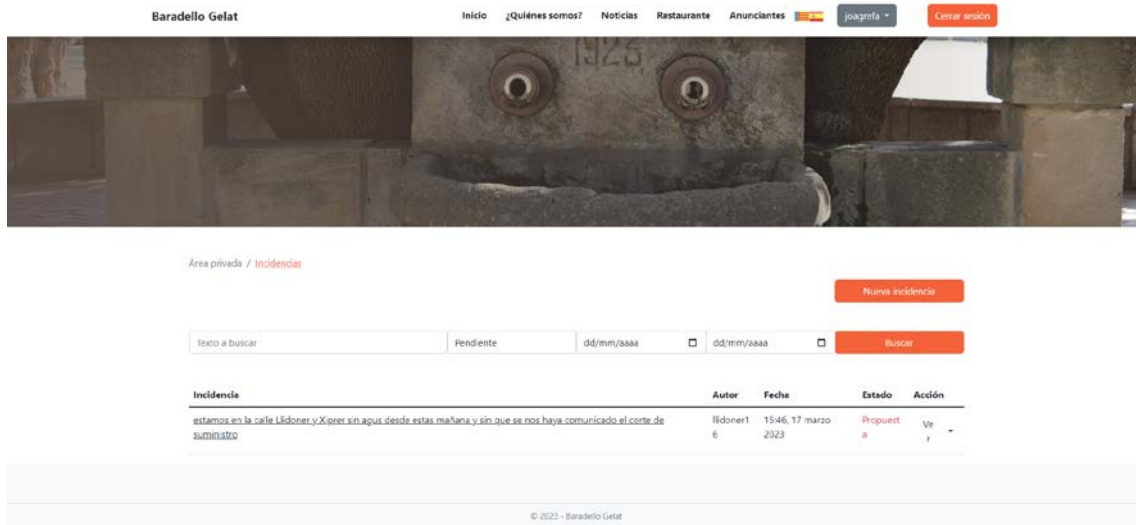


En els subforos, podrà afegir temes i escriure comentaris.

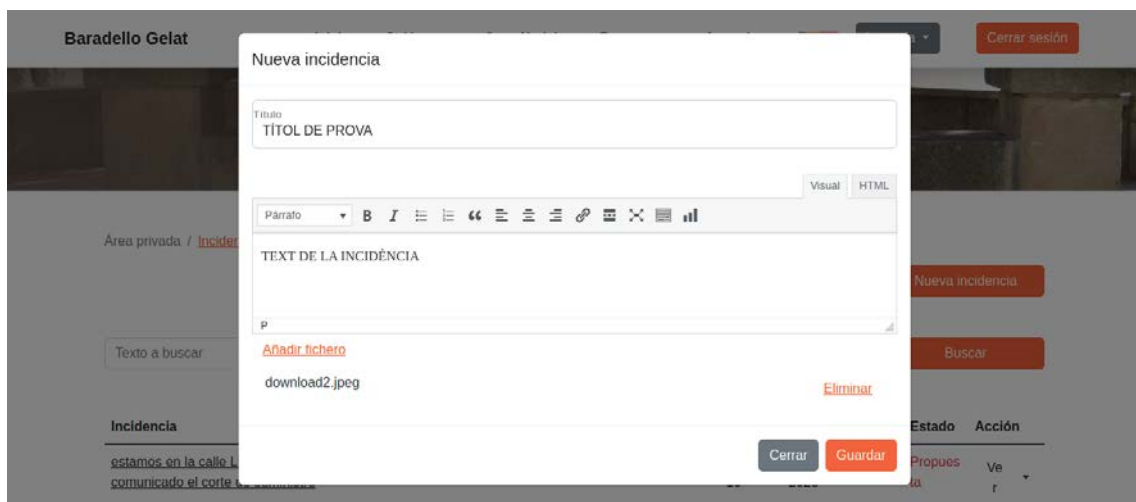




En la pàgina de les incidències, si és un administrador, les vorà totes, si no, sols li apareixeran les seues incidències desplegadas. En este cas, és la vista d'un administrador.



Si volem afegir una nova incidència, tindrem la vista següent.





Al guardar-la apareixerà com una nova incidència proposta.

Texto a buscar    Pendiente    mm/dd/yyyy    mm/dd/yyyy    **Buscar**

Incidencia	Autor	Fecha	Estado	Acción
<a href="#">TÍTOL DE PROVA</a>	joagrefa	23:24, 3 julio 2023	Propuesta	Ver
<a href="#">estamos en la calle Llidoner y Xiprer sin agus desde estas mañana y sin que se nos haya comunicado el corte de suministro</a>	llidoner 16	15:46, 17 marzo 2023	Propuesta	Ver

Si li donem a vore, anirem a la incidència on es vorà tot el text complet i els fitxers adjunts.

Área privada / [Incidencias](#) / TÍTOL DE PROVA

Propuesta

## TÍTOL DE PROVA

Joanma

03/07/2023

Acción

TEXT DE LA INCIDÈNCIA

### Adjuntos



download2

Acción


També se li podran afegir comentaris.

### Deja un Comentario

Empieza a escribir...

Publicar Comentario

Els administradors i membres de manteniment poden canviar-li l'estat a les incidències.

Texto a buscar	Pendiente	mm/dd/yyyy	📅	mm/dd/yyyy	📅	Buscar
Incidencia	Autor	Fecha	Estado	Acción		
<u>TÍTOL DE PROVA</u>	joagrefa	23:24, 3 julio 2023	Propuesta	Ver 		
<u>estamos en la calle Llidoner y Xiprer sin agus desde estas mañana y sin que se nos haya comunicado el corte de suministro</u>	llidoner 16	15:46, 17 marzo 2023	Aceptar Rechazar Cerrar Eliminar			

Primer, s'accepta i se li canvia l'estat.

Incidencia	Autor	Fecha	Estado	Acción
<u>TÍTOL DE PROVA</u>	joagrefa	23:24, 3 julio 2023	Aceptada	Ver 

Si es tanca la incidència, s'ha de canviar el filtre i filtrar per Totes i per data per a trobar-les.

Texto a buscar	Todas	06/06/2023	07/31/2023	Buscar
Incidencia	Autor	Fecha	Estado	Acción
<a href="#">TÍTOL DE PROVA</a>	joagrefa	23:24, 3 julio 2023	Cerrada	Ver ▾

També es pot rebutjar i la incidència estaria a l'espera que algú la torne a acceptar.

Texto a buscar	Todas	05/31/2023	07/11/2023	Buscar
Incidencia	Autor	Fecha	Estado	Acción
<a href="#">TÍTOL DE PROVA</a>	joagrefa	23:24, 3 julio 2023	Rechazada	Ver ▾

## 5. Conclusió i treballs futurs

En conclusió, en este treball s'ha confrontat amb un problema real i uns requeriments d'un grup de persones per millorar el seu dia a dia. En general s'han complert tots els requisits proposats i s'ha desenvolupat una pàgina web usable per a qualsevol persona.

Des del punt de vista tècnic i professional, ha sigut un treball on he après, tant aspectes generals com concrets, sobre diferents tecnologies com WordPress, MySQL, Docker, Caddy, Linux, PHP, HTML, BootStrap, etc. Des del punt de vista humà, ha sigut un plaer i una experiència haver de contactar amb la gent de la urbanització per definir els requisits i fer el seguiment dels avanços. Des d'un punt de vista més general, estic molt content de l'experiència i els coneixements adquirits durant la realització d'este projecte i, a més, estic completament segur que esta experiència em servirà pel meu futur, tant personalment com professionalment. No obstant, no sols és un projecte que m'ha servit a mi, estic segur que els lectors d'este projecte aprendran també.

En quant als treballs futurs, tot projecte és millorable en molts aspectes. Es pot millorar en quant a usabilitat: Encara que és un projecte molt intuïtiu, hi ha parts que es podrien modificar. També es pot millorar en funcionalitat: Es va parlar d'afegir altres funcionalitats com un apartat paregut a BlaBlaCar per a que la gent avise quan va al poble i evitar viatges de més o un apartat amb un calendari per reservar la sala comú que té l'associació. També es pot millorar en quant a eficiència: Es podria optimitzar per visualitzar més fluidament l'aplicació.

Altre punt molt interessant seria desenvolupar una aplicació web/mòbil amb més funcionalitats i utilitzar l'API de WordPress, que és un servei que ofereix diferents possibilitats als programadors.

## 6. Bibliografia

- [1] Docker <https://docs.docker.com/>
  
- [2] WordPress for Web Developers. Autora Stephanie Leary.
  
- [3] Building Web Apps with WordPress. Autors Brian Messenlehner, Jason Coleman.  
Pròleg per Brad Williams.
  
- [4] Professional WordPress Design and Development. Autors Brad Williams, David  
Damstra, Hal Stern.
  
- [5] Bootstrap. <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
  
- [6] Caddy <https://caddyserver.com/docs/>
  
- [7] Caddyfile <https://caddyserver.com/docs/caddyfile>
  
- [8] Git <https://git-scm.com/docs>
  
- [9] YAML. <https://es.wikipedia.org/wiki/YAML>
  
- [10] Docker Compose <https://docs.docker.com/compose/>
  
- [11] Jerarquia plantilles WordPress.  
<https://developer.wordpress.org/themes/basics/template-hierarchy/>
  
- [12] Funcions WordPress <https://developer.wordpress.org/reference/functions/>

[13] Bash <https://www.gnu.org/software/bash/manual/bash.html>

[14] Escàner de vulnerabilitats 1. <https://github.com/projectdiscovery/nuclei>

[15] Escàner de vulnerabilitats 2. <https://github.com/gocaio/sct/releases/tag/v1.3.0>

[16] Atac XSS. <https://owasp.org/www-community/attacks/xss/>