

DESARROLLO DE APLICACIONES WEB

Segundo curso



Grupo: Marc Nicolás + Aitor Rodríguez

Plazo de entrega: 12 de mayo, 2024

Curso académico 2023 / 2024

Resumen

Este documento contiene el transcurso del proyecto en parejas de realizar desde cero una webapp, en nuestro caso una red social en formato de foro muy similar al del sitio Reddit.

El plazo total es de 2 meses de duración y a diferencia del proyecto anterior en este se busca enfatizar el uso de frameworks como Laravel10 + Vue 3 y Bootstrap para el diseño del frontend y herramientas como Yup, Eloquent, Axios y JavaScript para la implementación de funcionalidades adicionales.

Algunos de los puntos clave en los que enfatizaremos serán los objetivos, el apartado de diseño y por supuesto el de funcionalidad; como se entrelazan y la importancia de cada uno de estos aspectos como conjunto una vez finalizado el proyecto.

Palabras clave

RRSS, Laravel, Vue, Framework, Github, Coworking, Model, View, Controller, MVC, PHP, JavaScript, OOP, MySQL, HTML, CSS, Figma, Moodboard, Wireframe, Diagram, XAMPP, Bootstrap, UX/UI, GIT, Axios, Usuarios, Posts, Foro, Posts, Comentarios, Categorías, Accounts, Yup, PrimeVue, Eloquent, Validation forms, Objects...

Keywords

RRSS, Laravel, Vue, Framework, Github, Coworking, Model, View, Controller, MVC, PHP, JavaScript, OOP, MySQL, HTML, CSS, Figma, Moodboard, Wireframe, Diagram, XAMPP, Bootstrap, UX/UI, GIT, Axios, Usuarios, Posts, Foro, Posts, Comentarios, Categorías, Accounts, Yup, PrimeVue, Eloquent, Validation forms, Objects...

ÍNDICE

1 Introducción	4
2 Contexto y motivación del proyecto	5
2.1 Objetivos del proyecto.....	5
2.1.1 Objetivos generales de producción.....	5
2.1.2 Objetivos de aprendizaje	7
3 Descripción	8
3.1 Resultados esperados vs obtenidos.....	8
4 Análisis.....	9
4.1 Requisitos funcionales.....	9
4.2 Requisitos no funcionales	11
5 Requisito de datos.....	14
5.1 Identidad	14
5.1.1 Definición de la empresa	14
5.1.2 Identificación del target.....	15
5.1.3 Identificación de la imagen corporativa de la empresa	16
5.1.4 Identidad visual.....	17
5.2 Diseño visual y estructural.....	21
5.2.1 Diseño de la interfaz	21
5.2.2 Prototipo web.....	24
5.2.3 Diseño web final	27
5.3 Diseño de la base de datos.....	31
5.3.1 Modelo entidad-relación	31
5.3.2 Modelo relacional	32
5.4 Diagrama de clases / Diagrama UML.....	33
5.5 Diseño del sistema	34
6 Problemas y/o dificultades	36
7 Conclusión.....	38
8 Referencias	39
9 Guía de evaluación	40

1 Introducción

Proyecto principal donde se realizará una aplicación web basada en una red social similar a Reddit, permitiendo a los usuarios publicar contenido, votar y comentar en publicaciones.

Funcionalidades base:

- El sitio web de simulará la creación de un aplicativo web que permita a los usuarios poder crear publicaciones propias e interactuar con las de otros.
- Los usuarios podrán registrarse y hacer uso de su cuenta a lo largo del sitio.
- Se implementa un sistema de *upvote-downvote* y en función a esto el contenido aparecerá más arriba o más abajo de la lista según el tipo de filtro aplicado.

Funcionalidades distintivas.

- El texto de las publicaciones y comentarios están limitados a 300 caracteres.
- Un filtro para los posts peor valorados. (mayor cantidad de downvotes)
- Durante el registro se realiza un test (de personalidad) de 3 preguntas que definirán la foto de perfil del usuario al azar y esta no podrá ser reemplazada.

A pesar de que algunas de las funcionalidades serán similares, otras se adaptaran para poder cumplir con la carga de trabajo dentro del plazo del proyecto, y por otro lado se implementaran otras de distintivas.

** Cabe decir que los requisitos obligatorios del proyecto tienen una prioridad sobre las funcionalidades distintivas.*

2 Contexto y motivación del proyecto

2.1 Objetivos del proyecto

2.1.1 Objetivos generales de producción

Una fase inicial decidimos partir del diseño de webs de redes sociales, adaptándolo a las funcionalidades que queremos implementar e incluso en muchos casos dotando de un tono minimalista para reducir la carga de trabajo y optimizar el sitio web.

Organización y seguimiento de las tareas:

- Trello – Organizar tareas y tiempos
- Discord – Chat de texto y voz, compartición de documentos

Realizar diseños de forma fluida y concisa mediante plataformas específicas para ello:

- Affinity Designer – Moodboard
- Figma – Wireframe
- Figma – Maquetaje web

Diseñar e implementar de una BBDD con tal de poder realizar las funcionalidades.

- Modelo Entidad/Relación
- Modelo Relacional
- MySQL – Estructura de base de datos

Preparamos el entorno de desarrollo para facilitar la siguiente fase.

- Visual Studio Code - Configuración específica del IDE (framework) + composer
- GitHub – Desarrollo del código en la nube (coworking)
- XAMPP – Hospedaje local de la página web y la BBDD

Una segunda fase basada en el desarrollo teniendo en cuenta todo lo ya realizado previamente, trabajando en equipo basada implementar el diseño y funcionalidades al producto final.

Implementar las funcionalidades.

- Laravel 10 – Framework PHP para backend
- VueJS 3 – Framework JavaScript para frontend
- Axios – Librería de JavaScript

Implementar el diseño.

- HTML5 – Estructura web
- Bootstrap 5.2 – Diseño de los bloques y a nivel de organización
- CSS3 – Diseño a nivel de aspecto
- Análisis heurístico de Jakob Nielsen – Examen UX/UI

Hospedaje web.

- XAMPP – Servidor local

2.1.2 Objetivos de aprendizaje

Con este proyecto se busca adquirir, repasar y practicar los siguientes conceptos:

- Aprender el lenguaje de programación PHP.
- Uso y administración de sesiones con PHP.
- Trabajar el uso avanzado de cookies.
- Trabajar con el framework de Laravel10 (backend).
- Aprender el lenguaje de programación JavaScript.
- Trabajar con el framework de VueJS 3 (frontend).
- Implementar API de JavaScript.
- Utilizar diferentes librerías como por ejemplo Axios.
- Aplicación de conceptos de diseño web.
- Realizar maquetas web como por ejemplo *moodboard* o *wireframes*.
- Creación de interfaces web.
- Analizar e identificar errores de UX/UI.
- Utilizar los diferentes elementos de HTML y hojas de estilo.
- Implantación de Bootstrap.
- Combinación de PHP con HTML.
- Creación de código limpio y estructurado.
- Creación y utilización de clases PHP.
- Aprender, comprender y llevar a práctica los conceptos de abstracción polimorfismo y clases abstractas.
- Trabajar en equipo mediante sistemas de control de versiones distribuido; GIT.
- Documentación del código.

3 Descripción

3.1 Resultados esperados vs obtenidos

El resultado final esperado debe tener un diseño lo más cercano posible a la maqueta web realizada mediante Figma durante la fase inicial y siempre conservando las características más identificativas de la imagen de marca a crear.

También debe estar dotada de adaptabilidad en cuanto a la resolución de pantalla se refiere: relación de aspecto 16:9 (*desktop*) y 9:16 (*smartphone*).

En cuanto a funcionalidades podemos ver una diferencia entre el documento inicial y este:

Resultados esperados	Resultados obtenidos
Mostrar las publicaciones.	Mostrar las publicaciones.
Votar las publicaciones.	Votar las publicaciones.
Comentar en las publicaciones.	Comentar en las publicaciones.
Crear una publicación, con o sin imagen.	Crear una publicación, con o sin imagen.
~ Filtrar las publicaciones por categorías . →	~ Filtrar las publicaciones por votos .
Realizar el proceso de <i>sign up</i> para crear una cuenta.	Realizar el proceso de <i>sign up</i> para crear una cuenta.
Realizar proceso de <i>sign in</i> mediante unas credenciales almacenadas en la BBDD.	Realizar proceso de <i>sign in</i> mediante unas credenciales almacenadas en la BBDD.
Ver los datos de su cuenta.	Ver los datos de su cuenta.
Borrar la cuenta.	Borrar la cuenta.
	+ Acceder al perfil del propio usuario.
	+ Acceder al perfil de otro usuario.
	+ El usuario puede editar sus posts.
	+ El usuario puede borrar sus posts.
	+ Borrar comentarios.
	+ Modificar los datos de la cuenta.
	+ Admin: eliminar posts.
	+ Admin: eliminar comentarios.
	+ Admin: eliminar usuarios.
	+ Tema oscuro/claro.
	+ Foto de perfil aleatoria.

4 Análisis

4.1 Requisitos funcionales

RF01 - Autenticación y registro de usuarios Requerido

- Implementar un sistema de inicio de sesión y registro en PHP.
- Diseñar un formulario de inicio de sesión y registro con campos de usuario y contraseña.
- Utilizar credenciales del usuario previamente almacenadas en una BBDD.

RF02 - Listado de publicaciones Requerido

- Crear una vista para mostrar el listado de publicaciones.

RF03 - Lógica del contenido Requerido

- Permitir a los usuarios agregar publicaciones, comentarlas y votarlas.

RF04 - Utiliza MVC Requerido

- Organizar el código utilizando el patrón Modelo-Vista-Controlador (MVC).

RF05 - Correcta integración PHP, HTML y JS Requerido

- Integrar PHP, HTML y JS de manera coherente y organizada.
- Utilizar PHP para generar dinámicamente contenido HTML.
- Evitar la mezcla excesiva de lógica PHP y código HTML.

RF06 - Sistema de búsqueda y filtrado de datos Requerido

- Implementación de un sistema de búsqueda y filtrado de datos de dos tipos, con Eloquent y con Vue.

RF07 - Peticiones HTTP y los flujos de trabajo Requerido

- Creación de rutas y controladores para manejar las peticiones HTTP y los flujos de trabajo de la aplicación.

RF08 – Sistema de roles Requerido

- Implementación de un sistema de gestión de permisos y roles para los usuarios. La web deberá tener una zona a la que podrán acceder dependiendo de los permisos asignados al usuario.
- Solo los usuarios administradores podrán acceder al panel de administración.

RF09 - Uso de Git Requerido

- Inicializar un repositorio Git para el proyecto.
- Realizar commits regulares para rastrear cambios en el código.
- Utilización de GIT durante todo el proyecto para trabajar en equipo. Además, al finalizar el proyecto, el repositorio de GIT debe contener toda la información y documentación necesaria para su despliegue (manual de instalación en un readme en GIT).

RF10 - Uso de Eloquent en Laravel Requerido

- Integración con una base de datos MySQL mediante el uso de Eloquent en Laravel.
- Gestionar la BBDD desde el IDE, realizar migraciones, crear tablas, modificar, etc.

RF11 - Las publicaciones y comentarios se almacenan en la BBDD

- Cuando se realiza una publicación o comentario estos se almacenan en la base de datos.

RF12 - Contraseña encriptada

- Las contraseñas de los usuarios se almacenan de forma encriptada mediante hash.
**Por norma general Laravel encripta por defecto las contraseñas.*

RF13 - Código comentado

- El código dispone de comentarios que explican el funcionamiento de este de forma clara y concisa.

4.2 Requisitos no funcionales

RNF01 – README.md Requerido

- Elaborar un documento README.md en el proyecto (GIT) que englobe todos los modelos/controladores/vistas, sus métodos/funciones, ruta o relaciones entre estas si fuese relevante.
- Contiene información importante sobre el proyecto, como su propósito, instalación, configuración, uso, etc.
- El formato “Markdown” permite incluir texto formateado de manera sencilla, con títulos, listas, enlaces, etc.

RNF02 – Diagrama de clases (UML) Requerido

- Se debe crear un diagrama UML que represente visualmente la estructura, relaciones entre modelo y controlador.

RNF03 - Diagrama/s de la base de datos Requerido

- Se debe crear un diagrama de la base de datos que represente visualmente la estructura, relaciones y componentes de la base de datos.
- Modelo entidad-relación.
- Modelo relacional.

RNF04 - Buen diseño de la base de datos Requerido

- El diseño de la base de datos debe ser eficiente, considerando la normalización para reducir redundancias y mejorar la integridad de los datos.
- Deberá contemplar claves primarias y foráneas claras, y documentación detallada del esquema de relaciones.
- Diseño y creación de una base de datos adecuada y eficiente, con un esquema de tablas bien estructurado y relaciones entre ellas. Deberá contar con tres relaciones N:M. Como mínimo, una de ellas debe tener campos en la tabla que genera la relación que no sean claves primarias.

RNF05 - Header (diseño) Requerido

- El diseño de la sección superior de la página debe ser lo más fiel a la maqueta propuesta.
- Debe disponer de una barra de navegación, aunque esta no se implemente ninguna funcionalidad a esta.
- El logo y los iconos deben estar en formato SVG.

RNF06 - Footer (diseño) Requerido

- El diseño de la sección inferior de la página debe ser lo más fiel a la maqueta propuesta.
- Debe incluir iconos de redes sociales, a poder ser en formato SVG.
- Constará el nombre de ambos alumnos.

RNF07 - Página de inicio (diseño) Requerido

- El diseño de la página principal debe ser lo más fiel a la maqueta propuesta.
- Debe permitir al usuario navegar de forma fácil al resto de páginas.

RNF08 - Página de inicio de sesión (diseño) Requerido

- El diseño debe ser lo más fiel a la maqueta propuesta.
- Debe permitir al usuario iniciar sesión de forma sencilla e inequívoca.

RNF09 - Página de registro (diseño) Requerido

- El diseño de la cesta debe ser lo más fiel a la maqueta propuesta.
- Debe permitir al usuario crear una cuenta de forma sencilla e inequívoca.

RNF10 - Archivos CSS Requerido

- Crear archivos de estilo CSS para dar estilo y diseño coherente a todas las páginas.
- Utilizar reglas CSS para aplicar estilos específicos a cada componente y página.

RNF11 - Responsive para Smartphones Requerido

- Garantizar que todas las páginas y componentes sean responsive, es decir, su aspecto debe adaptarse a diferentes ratios de relación de aspecto: 16:9 (*desktop*) y 9:16 (*smartphone*).

RNF12 - Imágenes con ALT

- Todas las imágenes cuentan con ALT para garantizar una correcta accesibilidad a los usuarios.

RNF13 - Resolución imágenes correcta

- La calidad de las imágenes debe ser la mayor posible.
- El peso de las imágenes debe tenerse en cuenta.

RNF14 - Uso de iconos en SVG

- Todos los iconos utilizados en la página deben estar en formato SVG.

RNF15 - Logo en SVG

- El logo de la página debe estar en formato SVG.

RNF16 - Uso adecuado de tipografías

- Las tipografías utilizadas en la página deben ser las adecuadas para cada contexto según la maqueta.
- Deben garantizar una correcta legibilidad.

RNF17 - Buena organización de las hojas de estilo

- Las hojas de estilo utilizadas deben estar correctamente ordenadas.
- Puede utilizarse una sola hoja o varias siempre que se garantice un orden adecuado.

RNF18 - Uso de Bootstrap, al menos del ROW y COL correctamente

- Se debe utilizar Bootstrap para formatear el diseño.
- Mínimo requisito, utilizar las propiedades de fila (ROW) y columna (COL).

RNF19 - Diseño igual que prototipo o mejorado

- El diseño debe ser fiel al prototipo/maqueta.
- En caso de realizar modificaciones se debe justificar.

5 Requisito de datos

5.1 Identidad

5.1.1 Definición de la empresa

Pelipper es una empresa innovadora de mensajería online que conecta a usuarios de todo el mundo de manera rápida y segura.

Inspirada en el modelo de Reddit, Pelipper ofrece una plataforma intuitiva donde los usuarios pueden enviar mensajes, compartir contenido y participar en comunidades temáticas.

Con un enfoque en la privacidad y la moderación efectiva, Pelipper garantiza una experiencia de usuario libre de spam y contenido inapropiado.

5.1.2 Identificación del target

El público objetivo de Pelipper son principalmente jóvenes de entre 16 y 50 años, ya que es el rango de edad donde las relaciones sociales dependen mayormente de mantener el contacto o simplemente por ocio. Es por ello por lo que buscan una plataforma de mensajería online donde entretenerse sin tener que pensar mucho, es por ello que dispone de una interfaz sencilla, con colores sencillos, una cantidad de páginas limitada y un formato vertical donde para visualizar el contenido solo hay que hacer scroll.

La gama de colores es amigable a la vista para que el usuario pueda navegar durante largos lapsos de tiempo sin sufrir cansancio, es por ello que debemos aprovechar esta oportunidad para incluir sutilmente nuestra imagen de marca, en este caso en forma de color **#EF324C** ya que la idea parte de Reddit el cual usa un tono naranja, el cual nos parece idílico en pequeñas dosis sin embargo queremos tener una identidad propia, es por ello que elegimos este color.

*(Originalmente iba a ser **#F9AD5B** pero por problemas con el contraste se reemplazó por el actual; el cual combina perfectamente tanto sobre tonos claros como oscuros).

Este se encuentra entre el **fucsia** y el **naranja** pues queremos posicionarnos entre dos psicologías del color, pero sin ser estresante o cargante.

Además, combina bien tanto en modos claros como oscuros.

Fucsia:

- Simboliza la creatividad, la originalidad y la individualidad. En marketing, representa diversión, emoción, sofisticación. En tecnología, transmite modernidad y audacia.
- En los colores de la misma gama más vivos es ideal para atraer la atención del visitante. Usado en exceso resulta cargante. Debe ser sutil.
- Se utiliza mucho en el sector del entretenimiento y de los productos juveniles.

Naranja:

- Se suele utilizar en entretenimiento. Es un buen color para determinados negocios de ocio.
- Debe usarse con moderación, ya que puede saturar o resultar estresante si se abusa de él.

5.1.3 Identificación de la imagen corporativa de la empresa

La identidad de la marca Pelipper se centra en ofrecer una experiencia sencilla e incluso si se permite; adictiva, entretenimiento como recompensa al menor esfuerzo; simplemente hacer scroll.

Es el modelo que se utiliza en la actualidad, podemos ver muchos ejemplos de gran éxito; Instagram, TikTok, Reddit, “Twitter”, etc.

Para ello utiliza una interfaz sencilla y sin distracciones con un formato vertical para facilitar la navegación, Pelipper garantiza que los usuarios puedan disfrutar del sin apenas esfuerzo o distracciones.

Además, se prioriza la privacidad y la moderación efectiva para una experiencia libre de spam y contenido inapropiado.

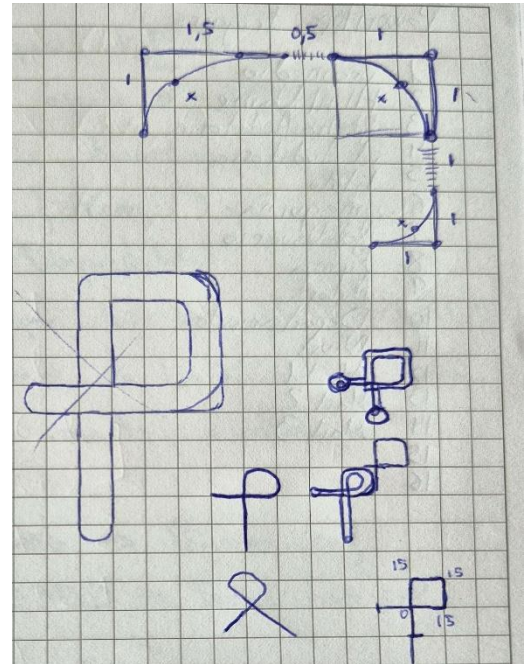
Con una paleta de colores amigable a la vista y un tono corporativo distintivo de lo que ya podemos encontrar busca destacarse mientras ofrece una identidad propia inspirada en la comodidad y la diversión.

La filosofía de la marca se basa en maximizar una fácil experiencia de uso para el usuario, evitando que tenga que aprender, llevándolo siempre de la mano y ofreciéndole el contenido en bandeja.

El nombre proviene de Pelipper, un “Pokemón” conocido por mucho público joven (dentro del target objetivo) y que a la vez representa un amigable pájaro mensajero.

5.1.4 Identidad visual

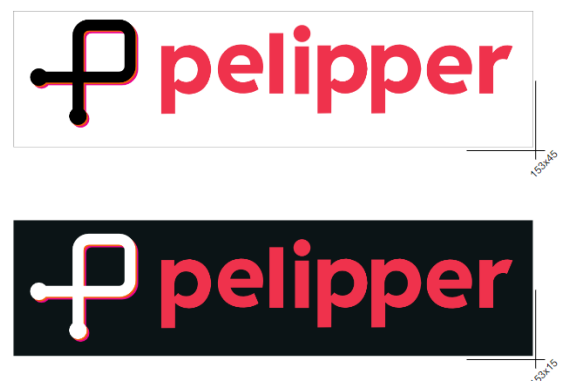
El logo ha ido evolucionando constantemente y sufriendo cambios radicales por el camino hasta el resultado actual.



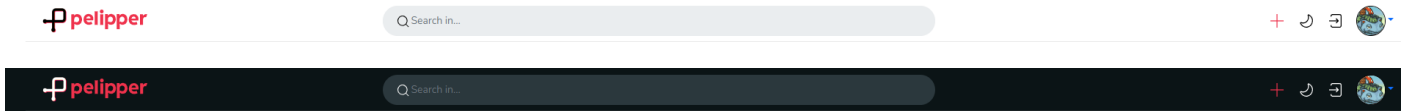
Logo



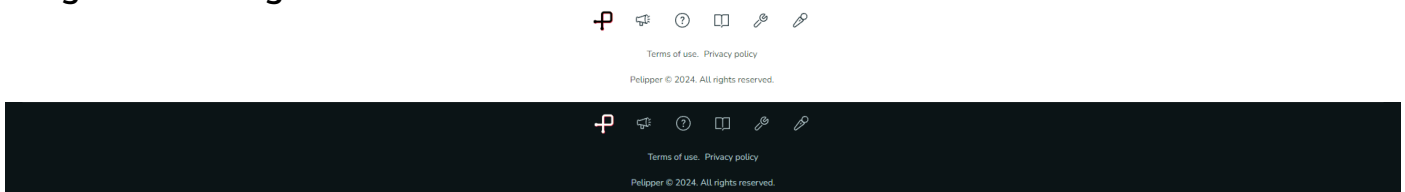
Modo de uso



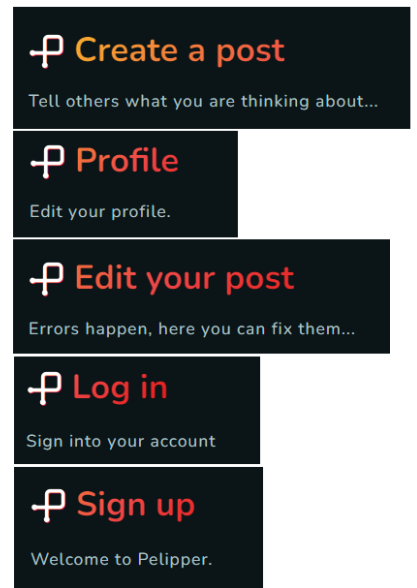
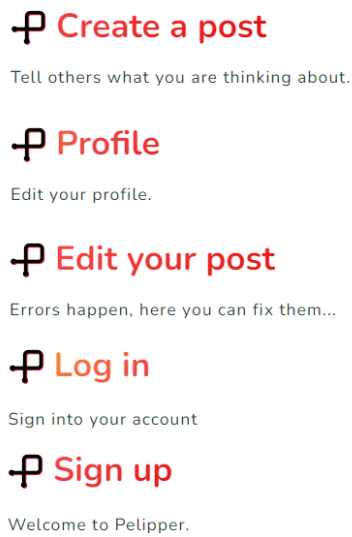
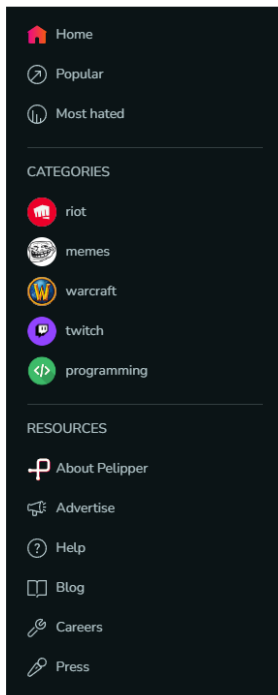
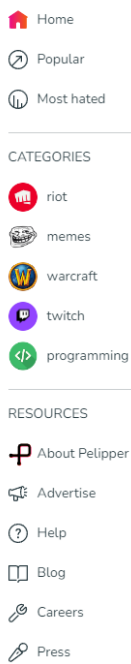
Integración del logo Header



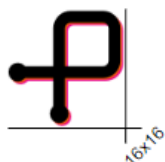
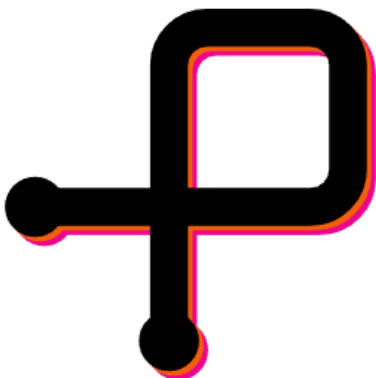
Integración del logo Footer



Integración del logo Otros

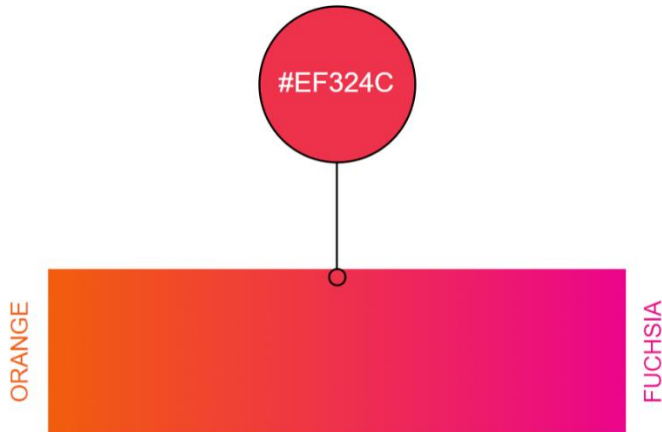


Integración del logo Favicon




Paleta de colores

Color de la marca obtenido justo del color intermedio entre naranja y fucsia, de tal forma que obtenemos un rosa “escarlata” que en algunos casos puede parecer incluso rojo.



Para ambos temas utilizamos una gama de colores totalmente contraria a la otra; en el claro hacemos uso de una gama de grises ligeramente azulados mientras que en el contrario intentamos prescindir del negro. No obstante, en el modo oscuro también se utilizan tonos azulados pero muy intensos para crear un contraste que no canse la vista.

BRAND COLOR		DARK MODE	LIGHT MODE	
	LOGO, BANNERS...	0B1416	FFFFFF	BACKGROUND
		0B1416	FFFFFF	CONTAINER BACKGROUND
		1A282D	F9FAFA	CONTAINER BACKGROUND:HOVER
		2B383D	EAEDF	BUTTON, SEARCHBAR
		223237	E2E7E9	BUTTON:HOVER, SEARCHBAR:HOVER
		B2C5C9	2A3C42	TEXT
		FFFFFF	000000	TITLE TEXT, ICON

Tipografía

Nunito Sans es una fuente sans-serif que forma parte del proyecto Google Fonts con lo cual es fácil de aplicar al proyecto.

Es una versión sans-serif de la fuente Nunito, diseñada por Vernon Adams.

Algunas de sus características son:

- Es una fuente moderna y amigable, con un estilo limpio y legible.
- Es adecuada para una amplia variedad de usos, incluyendo páginas web, aplicaciones móviles y diseños impresos.
- La familia de fuentes Nunito Sans incluye múltiples pesos, desde ligero hasta negrita, lo que la hace muy versátil para diferentes contextos de diseño.
- Ofrece una buena amplitud y espacio entre caracteres, lo que mejora la legibilidad, especialmente en tamaños más pequeños y en pantallas digitales.

Nunito Sans

1234567890

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890

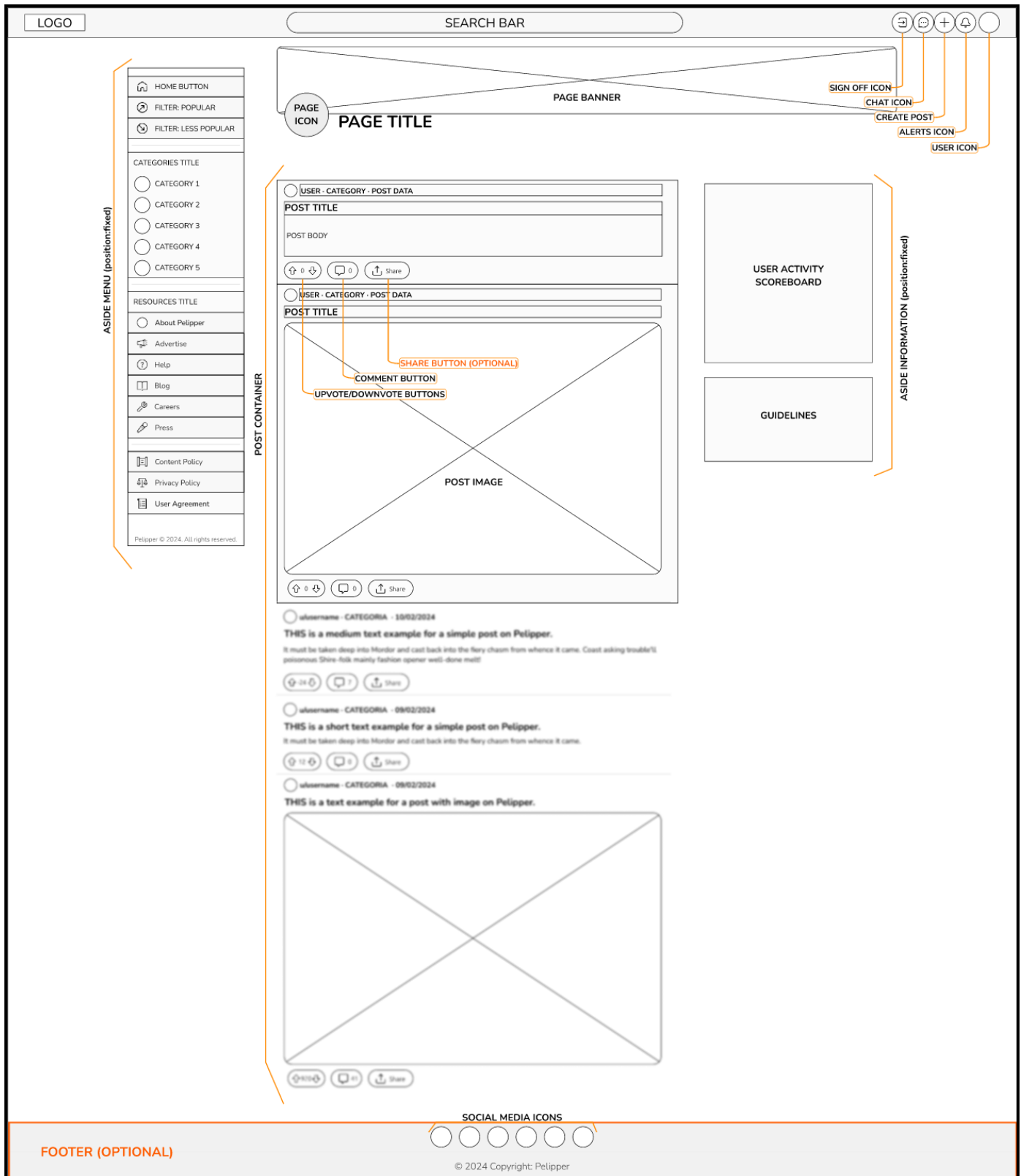
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

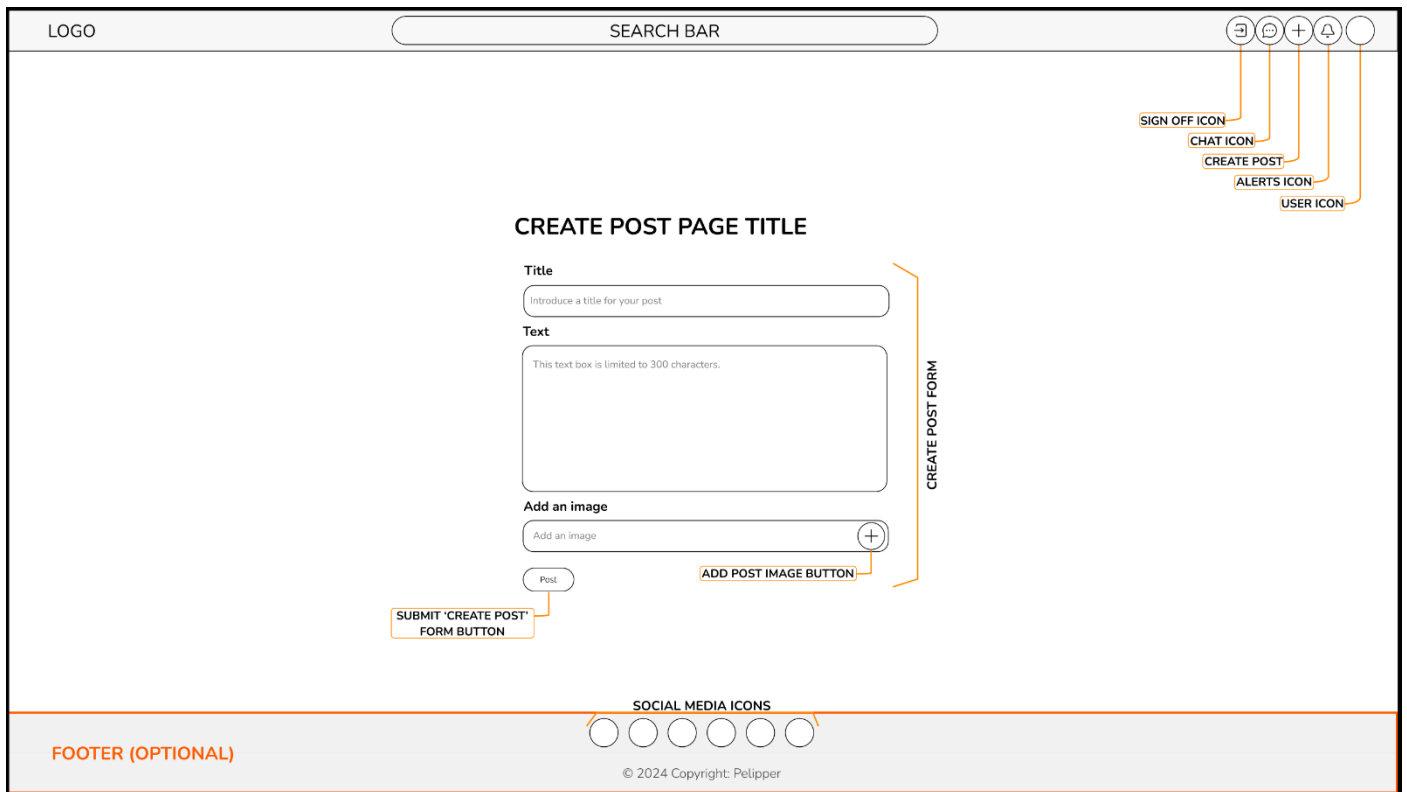
5.2 Diseño visual y estructural

5.2.1 Diseño de la interfaz

MAIN PAGE ([Ver en FIGMA](#))



CREATE POST



The wireframe illustrates a 'CREATE POST' page layout. At the top, a header bar contains a 'LOGO' placeholder, a 'SEARCH BAR', and a row of five icons: 'SIGN OFF ICON', 'CHAT ICON', 'CREATE POST' (highlighted), 'ALERTS ICON', and 'USER ICON'. The main content area is titled 'CREATE POST PAGE TITLE'. Below this, the 'CREATE POST FORM' is structured with three main sections: 'Title' (with a placeholder 'Introduce a title for your post'), 'Text' (with a placeholder 'This text box is limited to 300 characters.'), and 'Add an image' (with a placeholder 'Add an image' and a '+' button). A 'Post' button is located below the image section, with a label 'SUBMIT 'CREATE POST' FORM BUTTON' pointing to it. An 'ADD POST IMAGE BUTTON' label points to the '+' button in the image section. A vertical label 'CREATE POST FORM' is positioned to the right of the form fields. At the bottom, a 'FOOTER (OPTIONAL)' section contains 'SOCIAL MEDIA ICONS' (represented by six circles) and a copyright notice '© 2024 Copyright: Pelipper'.

INICIO DE SESIÓN

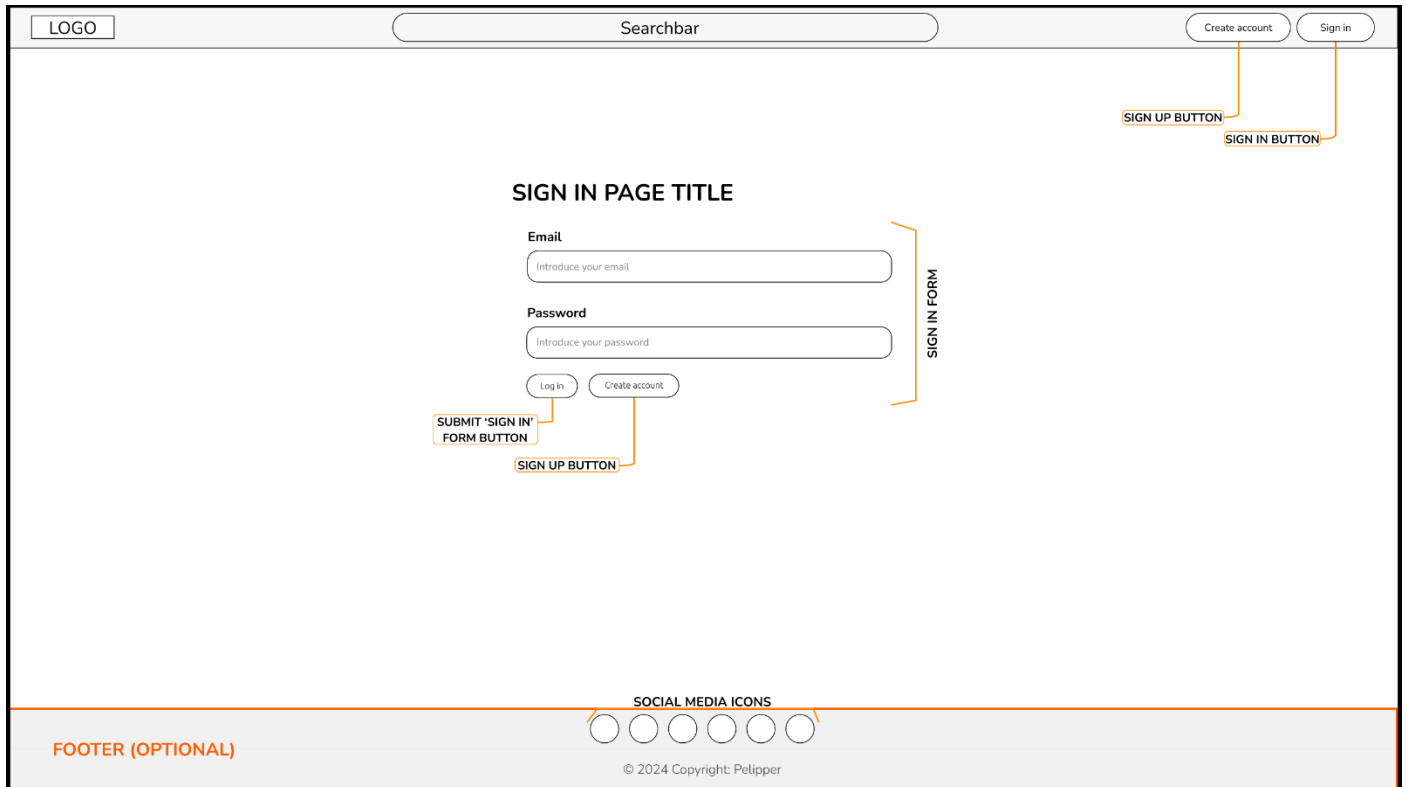


Diagram illustrating the Sign In page layout. The page features a header with a **LOGO**, a **Searchbar**, and buttons for **Create account** and **Sign in**. The main content area is titled **SIGN IN PAGE TITLE** and contains a form with fields for **Email** (placeholder: "Introduce your email") and **Password** (placeholder: "Introduce your password"). Below the password field are buttons for **Log in** and **Create account**. A bracket on the right side of the form is labeled **SIGN IN FORM**. A label **SUBMIT 'SIGN IN' FORM BUTTON** points to the **Log in** button. A label **SIGN UP BUTTON** points to the **Create account** button. The footer includes **SOCIAL MEDIA ICONS** and a **FOOTER (OPTIONAL)** section with the copyright notice: © 2024 Copyright: Pelipper.

REGISTRO

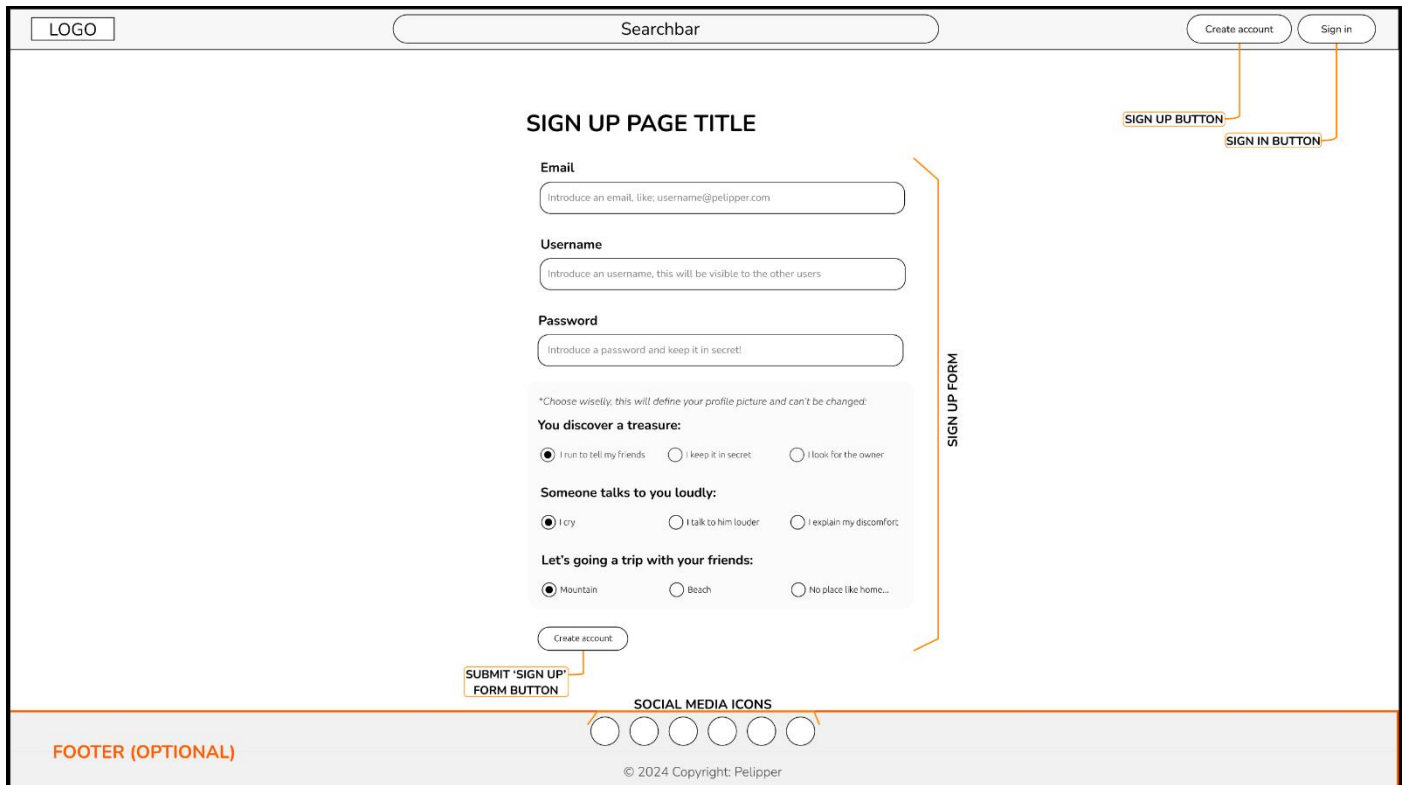
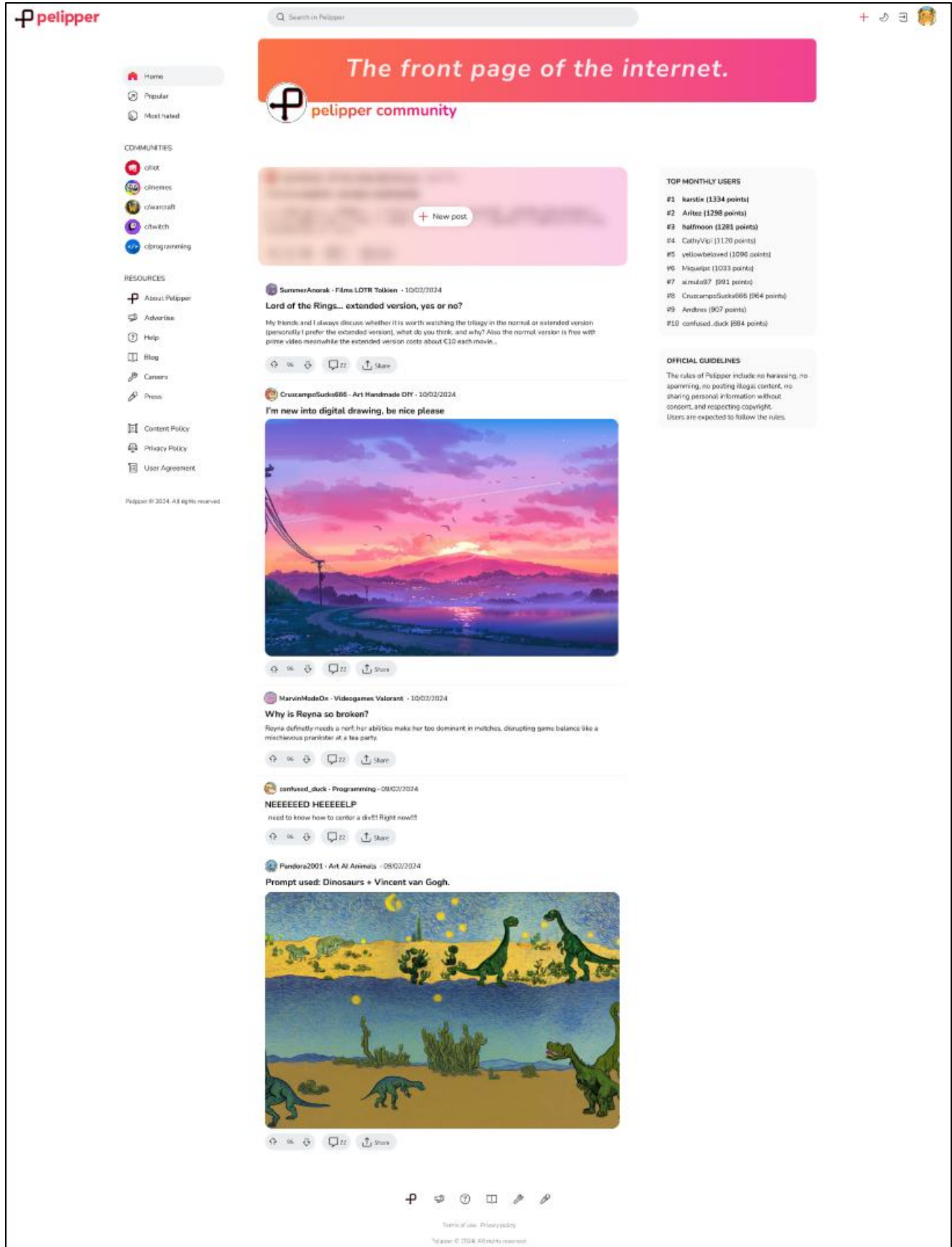



Diagram illustrating the Sign Up page layout. The page features a header with a **LOGO**, a **Searchbar**, and buttons for **Create account** and **Sign in**. The main content area is titled **SIGN UP PAGE TITLE** and contains a form with fields for **Email** (placeholder: "Introduce an email, like: username@pelipper.com"), **Username** (placeholder: "Introduce an username, this will be visible to the other users"), and **Password** (placeholder: "Introduce a password and keep it in secret!"). Below the password field is a section titled ***Choose wisely, this will define your profile picture and can't be changed:** with three sub-sections: **You discover a treasure:** (radio buttons: ☒ I run to tell my friends, ☐ I keep it in secret, ☐ I look for the owner), **Someone talks to you loudly:** (radio buttons: ☒ I cry, ☐ I talk to him louder, ☐ I explain my discomfort), and **Let's going a trip with your friends:** (radio buttons: ☒ Mountain, ☐ Beach, ☐ No place like home...). A bracket on the right side of the form is labeled **SIGN UP FORM**. A label **SUBMIT 'SIGN UP' FORM BUTTON** points to the **Create account** button. The footer includes **SOCIAL MEDIA ICONS** and a **FOOTER (OPTIONAL)** section with the copyright notice: © 2024 Copyright: Pelipper.

5.2.2 Prototipo web

MAIN PAGE




CREATE POST



Search in Pelipper

+ ↶ ↷ 🧑


 **Create a post**

Tell others what you are thinking about...

Title







Context

Context


Drop files here or [click here](#) to upload.

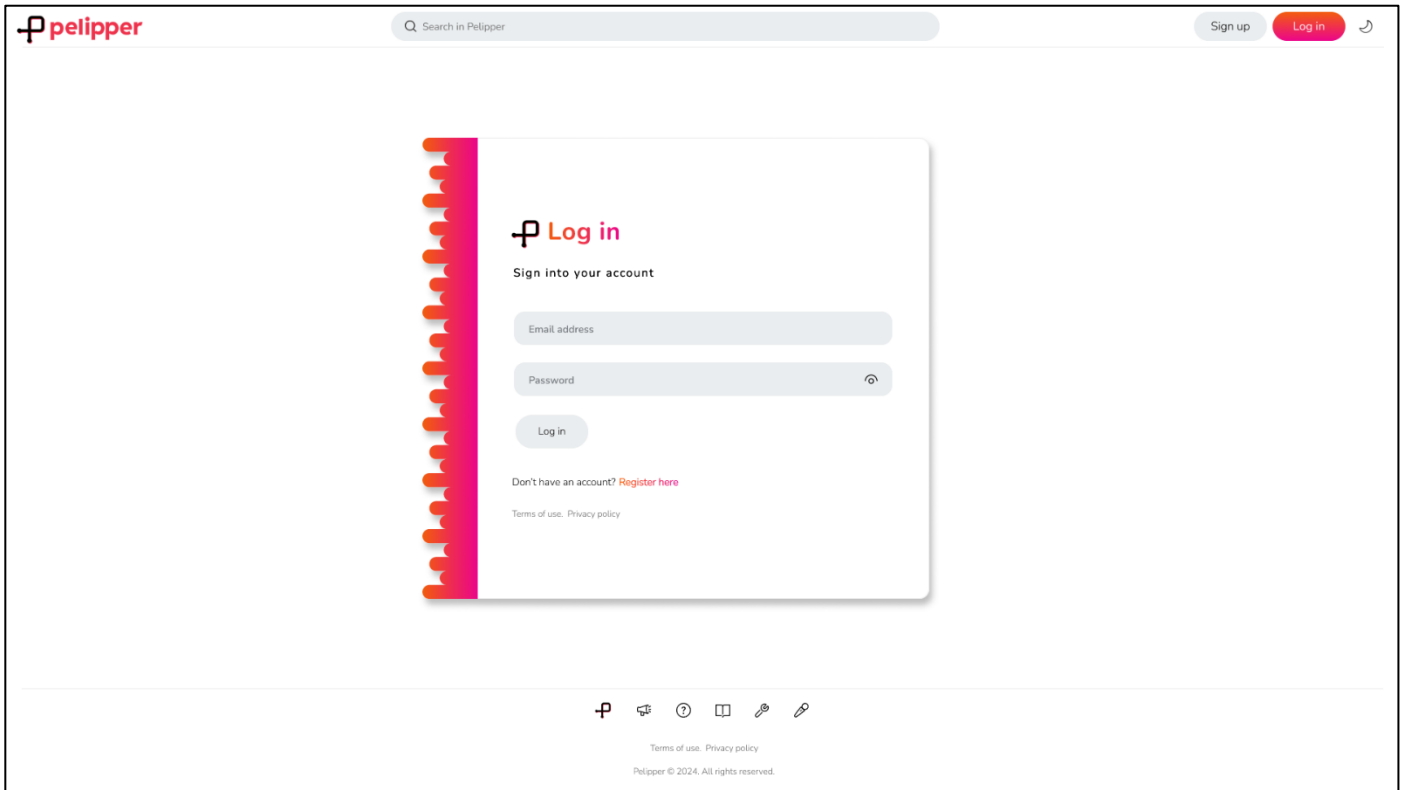
Post it

[Terms of use](#), [Privacy policy](#)



[Terms of use](#), [Privacy policy](#)
Pelipper © 2024. All rights reserved.

INICIO DE SESIÓN



The login form is centered on the page. It features a pink and orange decorative vertical bar on the left. The form includes a Pelipper logo, a title 'Log in', and a subtitle 'Sign into your account'. There are two input fields: 'Email address' and 'Password' (with an eye icon for toggling visibility). Below these is a 'Log in' button. At the bottom of the form, there is a link 'Don't have an account? Register here' and links for 'Terms of use' and 'Privacy policy'.

Search in Pelipper

Sign up Log in

Log in

Sign into your account

Email address

Password

Log in

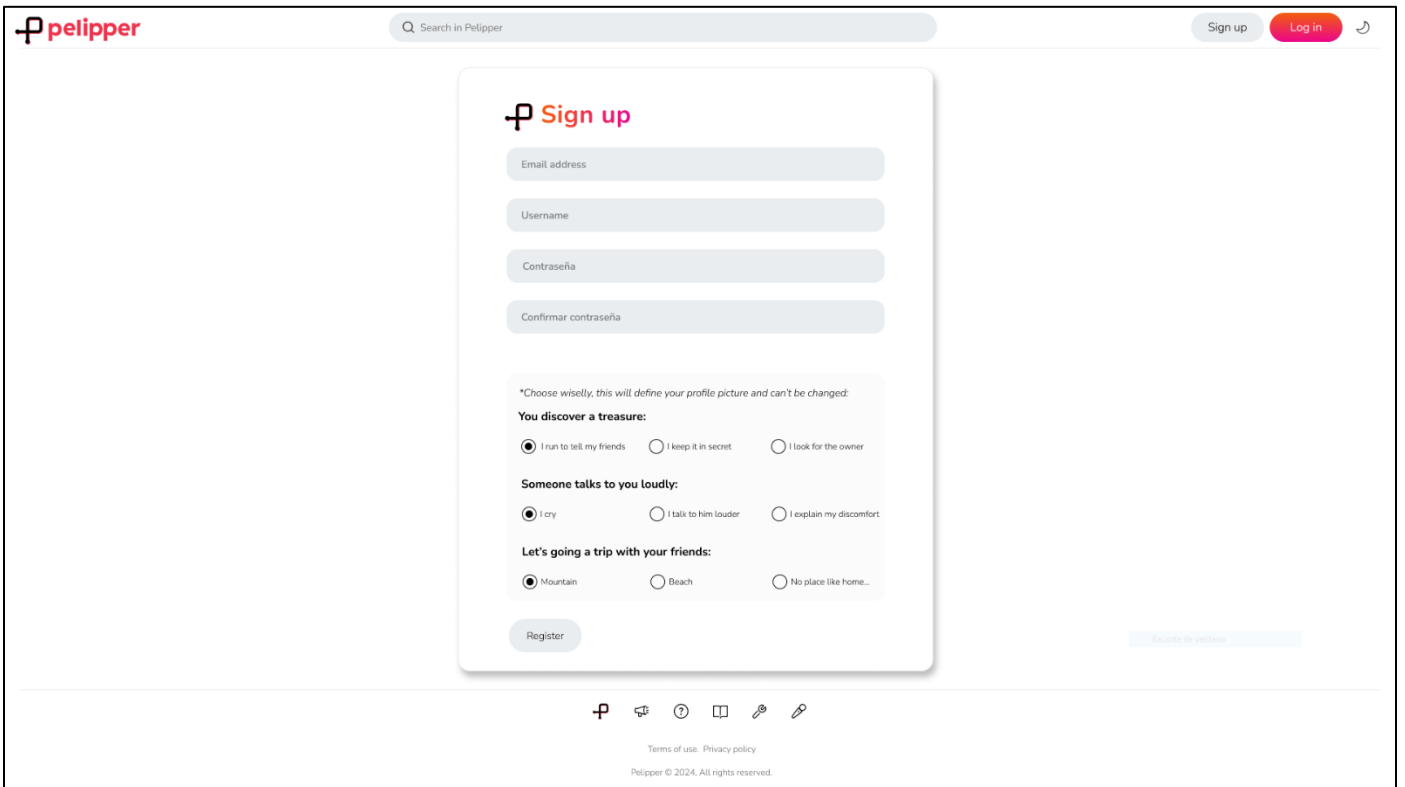
Don't have an account? [Register here](#)

[Terms of use](#) [Privacy policy](#)

Terms of use. Privacy policy

Pelipper © 2024. All rights reserved.

REGISTRO



The registration form is centered on the page. It features a Pelipper logo, a title 'Sign up', and a subtitle 'Create your account'. There are four input fields: 'Email address', 'Username', 'Contraseña', and 'Confirmar contraseña'. Below these is a 'Register' button. The form also includes a section for profile picture selection with three radio button options: 'I run to tell my friends', 'I keep it in secret', and 'I look for the owner'. There is also a section for 'Someone talks to you loudly' with three radio button options: 'I cry', 'I talk to him louder', and 'I explain my discomfort'. Finally, there is a section for 'Let's going a trip with your friends' with three radio button options: 'Mountain', 'Beach', and 'No place like home...'. At the bottom of the form, there is a link 'Recibir la verificación' and links for 'Terms of use' and 'Privacy policy'.

Search in Pelipper

Sign up Log in

Sign up

Create your account

Email address

Username

Contraseña

Confirmar contraseña

Register

*Choose wisely, this will define your profile picture and can't be changed:

You discover a treasure:

☒ I run to tell my friends ☐ I keep it in secret ☐ I look for the owner

Someone talks to you loudly:

☒ I cry ☐ I talk to him louder ☐ I explain my discomfort

Let's going a trip with your friends:

☒ Mountain ☐ Beach ☐ No place like home...

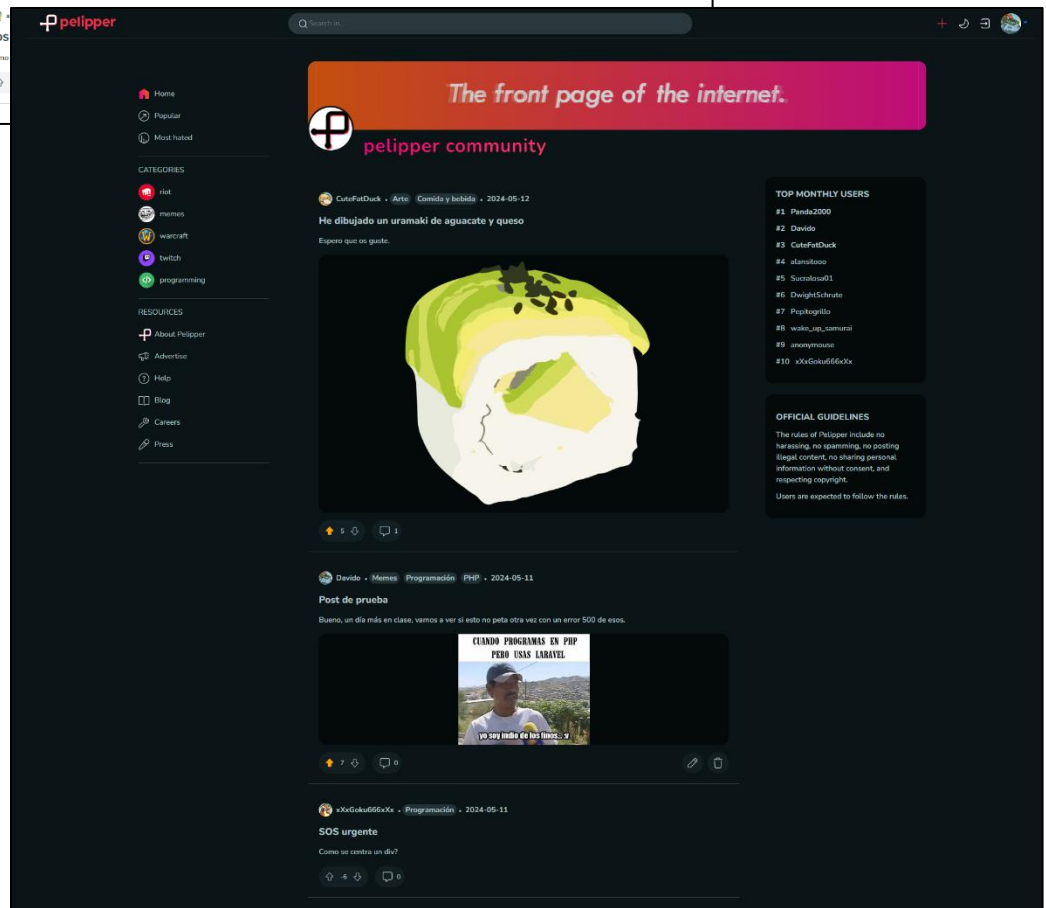
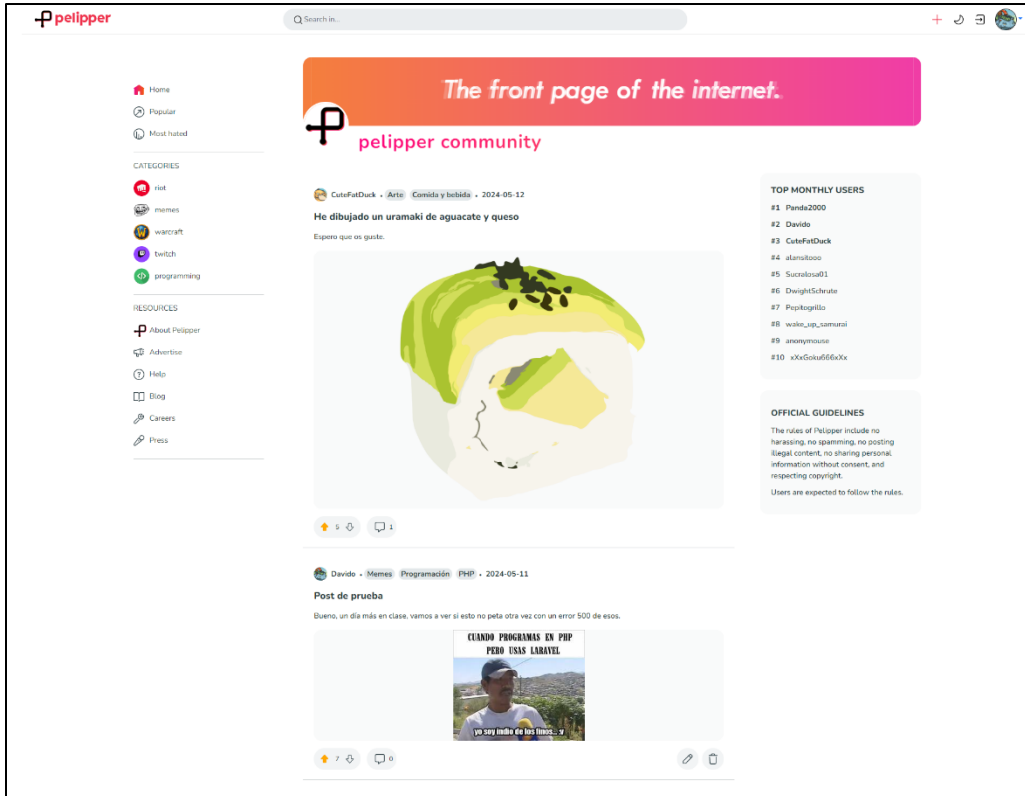
[Recibir la verificación](#)

Terms of use. Privacy policy


Pelipper © 2024. All rights reserved.


5.2.3 Diseño web final


MAIN PAGE




CREATE POST








Create a post


Tell others what you are thinking about...

Title
 Post de prueba


Context
 Bueno, un día más en clase, vamos a ver si esto no peta otra vez con un error 500 de esos.

Post it

Memes x Programación x PHP x

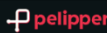



[Community guidelines](#) [Privacy policy](#)





[Terms of use](#) [Privacy policy](#)

Pelipper © 2024. All rights reserved.








Create a post


Tell others what you are thinking about...

Title
 Post de prueba

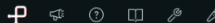
Context
 Bueno, un día más en clase, vamos a ver si esto no peta otra vez con un error 500 de esos.

Post it

Memes x Programación x PHP x




[Community guidelines](#) [Privacy policy](#)



[Terms of use](#) [Privacy policy](#)


Pelipper © 2024. All rights reserved.


INICIO DE SESIÓN



Q Search in...

Sign upLog in



 Log in

Sign into your account







Email address

Password

Log in


Don't have an account? [Register here](#)

[Terms of use](#) [Privacy policy](#)




[Terms of use](#) [Privacy policy](#)


Pelipper © 2024. All rights reserved.



Q Search in...

Sign upLog in



 Log in

Sign into your account







Email address

Password

Log in

Don't have an account? [Register here](#)


[Terms of use](#) [Privacy policy](#)




[Terms of use](#) [Privacy policy](#)

Pelipper © 2024. All rights reserved.

REGISTRO



Sign up
Log in



Sign up

Welcome to Pelipper.

*Choose wisely, this will define your profile picture and can't be changed:

You discover a treasure:

☐ I run to tell my friends
 ☐ I keep it in secret
 ☐ I look for the owner







Someone talks to you loudly:

☐ I cry
 ☐ I get angry
 ☐ I talk to him louder

Let's going a trip with your friends:


☐ Mountain
 ☐ Beach
 ☐ No place like home...

[Community guidelines.](#)
[Privacy policy](#)









[Terms of use.](#)
[Privacy policy](#)

Pelipper © 2024. All rights reserved.



Sign up
Log in



Sign up

Welcome to Pelipper.

*Choose wisely, this will define your profile picture and can't be changed:

You discover a treasure:

☒ I run to tell my friends
 ☐ I keep it in secret
 ☐ I look for the owner







Someone talks to you loudly:

☒ I cry
 ☐ I get angry
 ☐ I talk to him louder

Let's going a trip with your friends:

☒ Mountain
 ☐ Beach
 ☐ No place like home...

[Community guidelines.](#)
[Privacy policy](#)

[Terms of use.](#)
[Privacy policy](#)

Pelipper © 2024. All rights reserved.

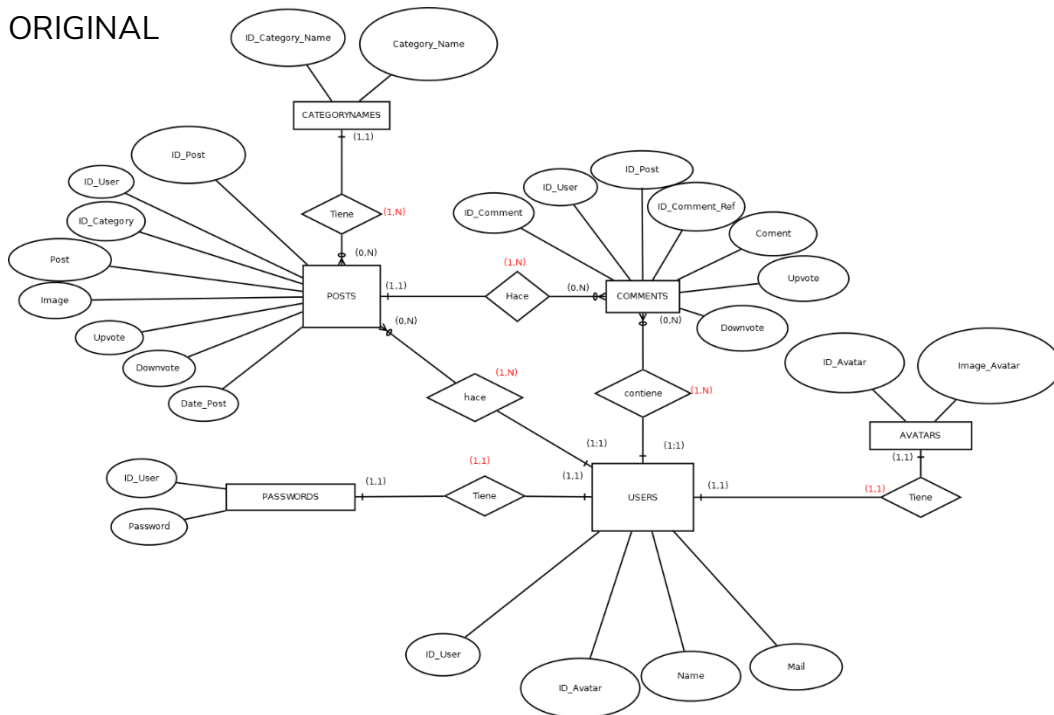
5.3 Diseño de la base de datos

5.3.1 Modelo entidad-relación

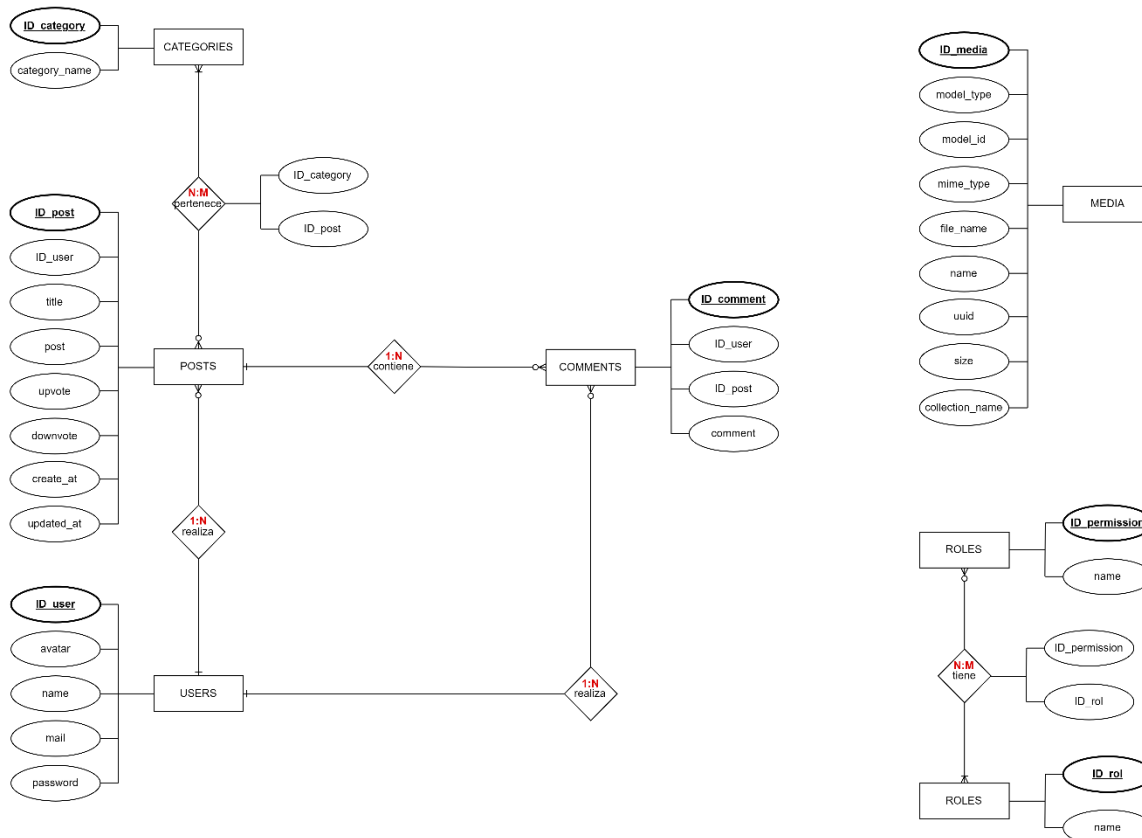
Pautas: https://docs.oracle.com/cd/A97335_02/apps.102/a81358/05_dev1.htm

*Puede variar de cara a la documentación final.

ORIGINAL



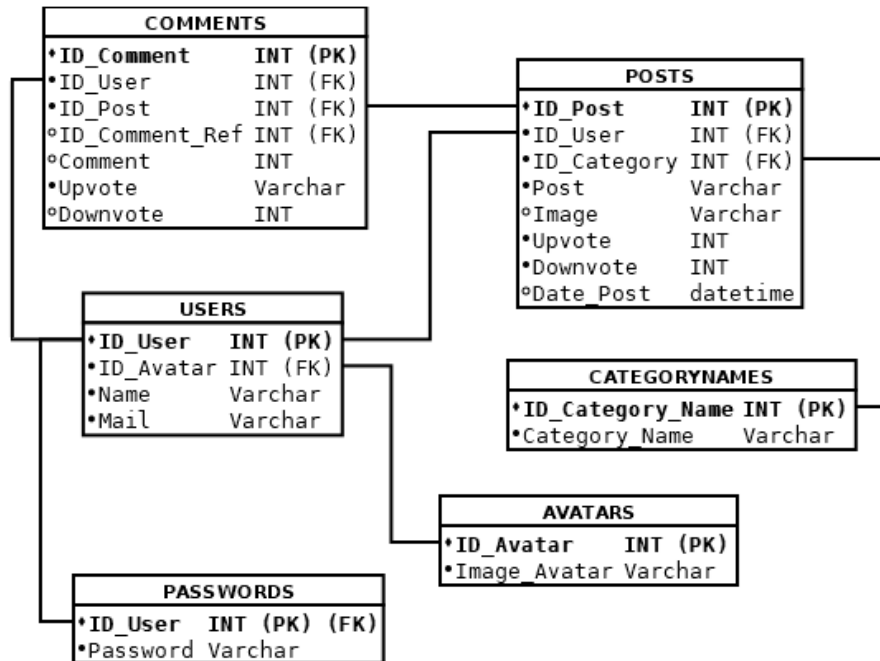
FINAL



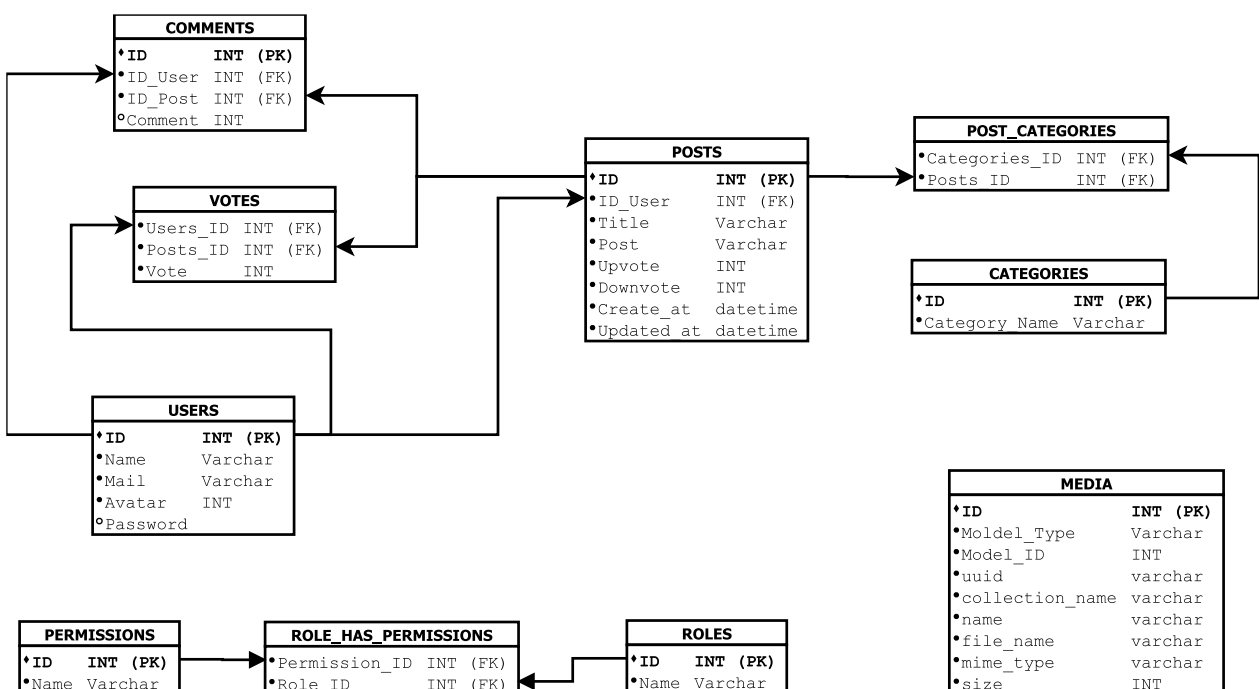
5.3.2 Modelo relacional

*Puede variar de cara a la documentación final.

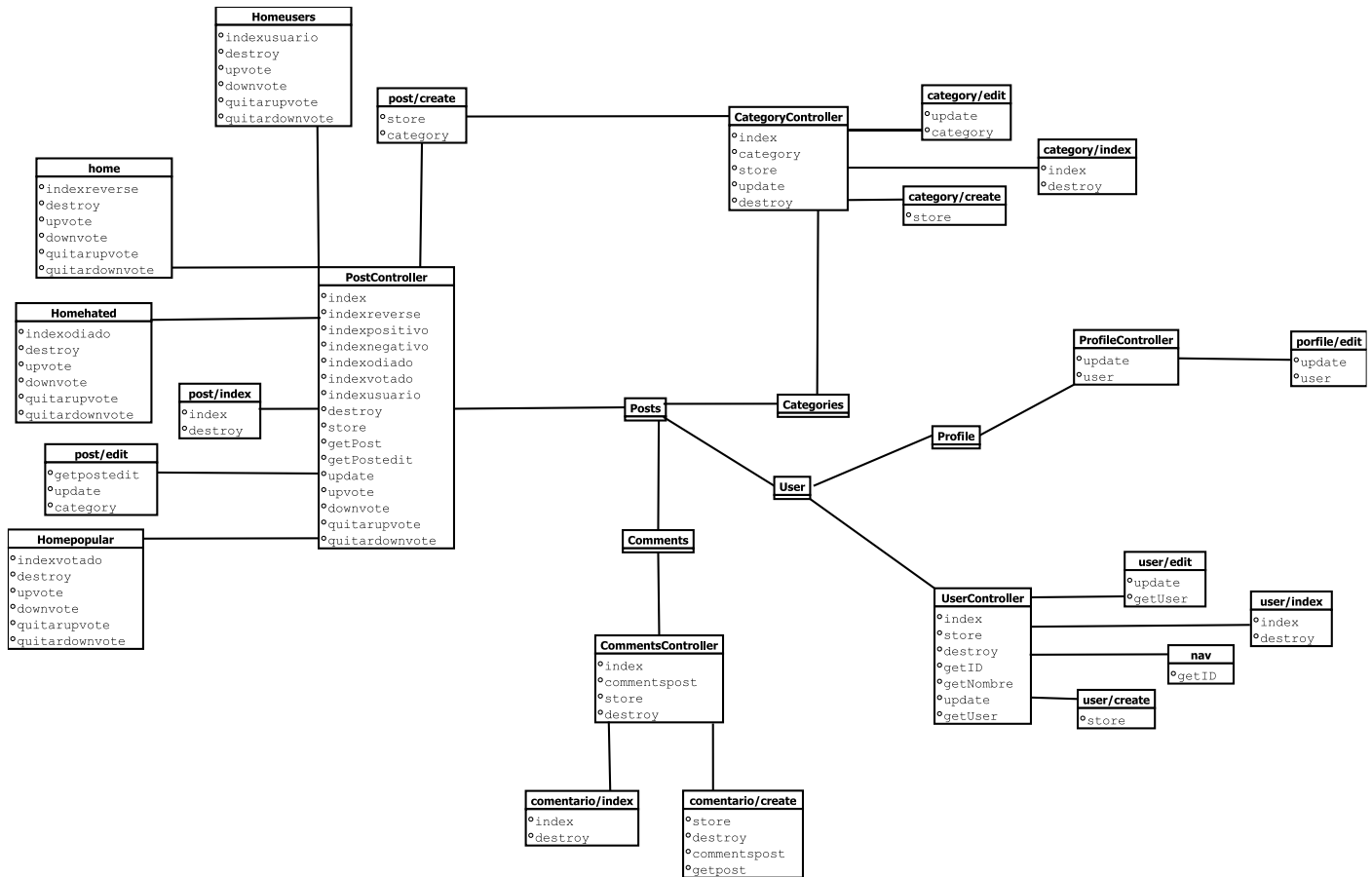
ORIGINAL



FINAL



5.4 Diagrama de clases / Diagrama UML

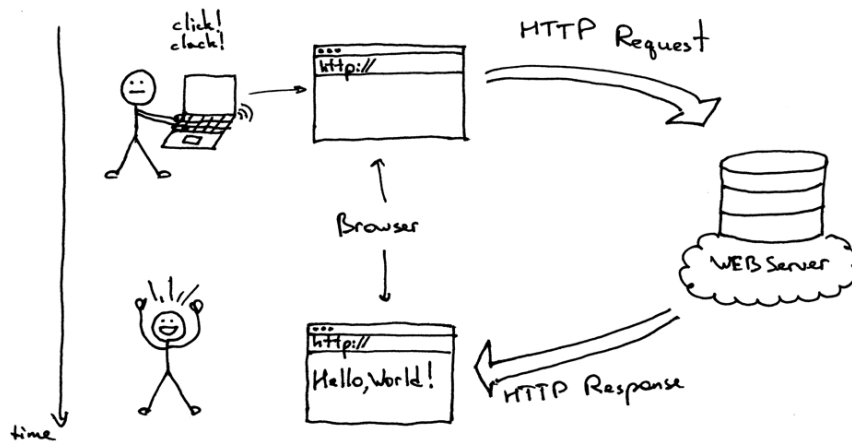
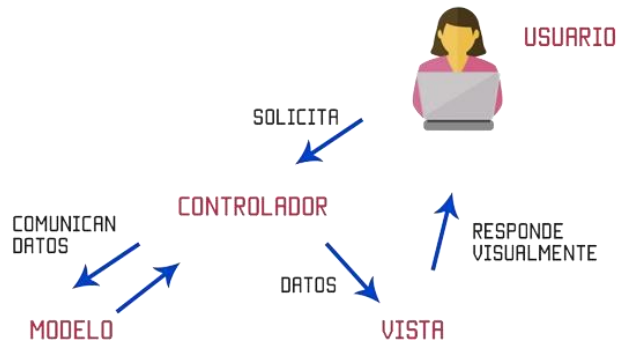


*Modelos: Posts, Comments, Categories, User, Profile

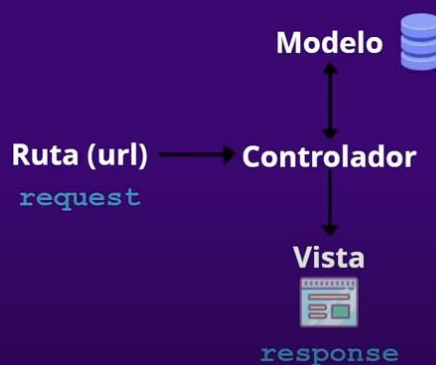
5.5 Diseño del sistema

Esquema del modelo cliente-servidor con las tecnologías

Tenemos una ruta la cual conecta con un controlador el cual puede consultar uno o más modelos o realizar consultas a la BBDD y finalmente retorna la vista con los datos en cuestión.



Dinámica básica Laravel



Tecnologías utilizadas:

- Hospedaje HTTP – XAMPP – Servidor Apache
- Hospedaje base de datos – XAMPP – Servidor MySQL
- Hospedaje de archivos – local
- Lenguaje de código – PHP(MVC) + CSS3(Bootstrap 5.3) + HTML5 + JS
- Desarrollo código – VisualStudioCode + GitHub
- Uso de frameworks – Laravel10 + VueJS3
- Uso de librerías – Notie.js + Axios...



6 Problemas y/o dificultades

1 - Gestionar upvotes y downvotes en la BBDD

El problema es que utilizábamos un update sobre la tabla, lo cual nos causaba problemas al realizar la función de votar positivo o negativo entre otras.

Solución: Para ello finalmente utilizamos método detach/attach/sync junto con una tabla pivot para gestionar relaciones de muchos a muchos en Laravel:

- El método detach elimina las relaciones existentes sin eliminar los registros relacionados en la tabla pivot.
- El método attach crea nuevas relaciones sin borrar las existentes, lo que puede causar duplicación de datos si no se controla.
- El método sync, por otro lado, sincroniza las relaciones eliminando las existentes y creando nuevas.

2 - Component: appfooter incluirlo solo en las vistas deseadas

Creemos un componente específico para implementar un footer en todas las vistas, el problema surgió a últimas horas del proyecto donde nos encontrábamos ciertas vistas que no requerían de este ya que eran de carácter vertical y ya integraban todo lo de este en un aside (por ejemplo, la *home page*).

Solución: Para ello integramos el componente manualmente en cada vista deseada, descartando a las que no.

3 - Editar los componentes de PrimeVue

A medida del proyecto hemos utilizado varios componentes del PrimeVue pero hemos encontrado que muchos de ellos tenían algunos aspectos estéticos que no sabíamos cómo editar.

Solución: Reducir el uso de componentes de PrimeVue, solo usar un par para probar. También eliminar el tema por defecto.

4 - Validar formularios (sin uso de Yup Vue)

Durante todo el proyecto hemos validado los formularios de forma “manual” al desconocer que era Yup, ahora ya es un poco tarde para ello, pero al menos nos hemos informado de que es y de como funciona. También hemos llegado a la conclusión de que hubiese sido útil utilizarlo, aunque ya no disponemos de tiempo para ello.

Solución: N/A.

5 – Problemas de responsive entre el ordenador de ambos compañeros

Aún utilizando una configuración de pantalla 1080p (1920x1080) con un ratio 16:9 y un aumento del 100% en ambos a uno se le ve de una forma y al otro de una muy distinta, así también el comportamiento de los contenedores, etiquetas, responsive; haciendo imposible llevar a cabo el desarrollo del diseño entre ambos equipos.

Solución: Desarrollar el diseño exclusivamente en uno de los dos ordenadores. Esto ha desbalanceado un poco la carga de trabajo en cuanto al diseño, pero hemos podido solventarlo presencialmente en clase.

6 – MySQL no inicia (XAMPP)

No hemos logrado descubrir a que se debe este error, pero se ha producido mínimo unas 3 veces a lo largo del proyecto, una vez aprendido a como reparar la BBDD desde los ficheros de configuración de XAMPP es cuestión de 5 minutos.

Solución: Reinstalar XAMPP y luego realizar un migrate con artisan sobre la BBDD de MySQL o reparar XAMPP.

[Video solución](#)

7 Conclusión

Durante el desarrollo de este proyecto utilizando Laravel 10 y Vue 3, hemos adquirido un profundo conocimiento sobre estas tecnologías y frameworks. Hemos aprendido a aprovechar al máximo las características y funcionalidades que ofrecen, lo que nos ha permitido construir una aplicación web robusta, eficiente y escalable.

Algunas de las lecciones más importantes que hemos aprendido son:

- La combinación de Laravel y Vue nos ha proporcionado una forma eficiente de construir aplicaciones web modernas y complejas, pero de forma sencilla, modular y escalable.
- Vue 3 nos ha impresionado con su sintaxis, mucho más clara y limpia que otras metodologías usadas en anteriores proyectos, sobre todo a la hora de integrar API's.
- Gestionar la BBDD desde el IDE gracias a Eloquent.
- Una vista sin código php intercalado de forma de manera poco estética.
- Desarrollar código mucha más eficientemente gracias al composer.
- Utilizar tabla pivot en casos específicos.

A pesar de los muchos obstáculos que nos hemos encontrado a lo largo del desarrollo, consideramos que el resultado final puede llegar a satisfacer con creces nuestro objetivo inicial.

Nuestro mayor problema quizá haya sido el tiempo pues ambos consideramos que hemos destinado más de un 50% del tiempo del proyecto a aprender a utilizar el nuevo entorno y cada una de sus particularidades. Otra gran dificultad o al menos así nos parece ya en gran retrospectiva es empezar a partir de un template ya elaborado en vez de un proyecto en blanco, es decir, desde cero.

En síntesis; consideramos este proyecto como algo muy fructífero para nuestra experiencia y curriculum ya que se trata de una tecnología fundamental hoy en día, llevando a otro nivel todo lo que hemos realizado antes durante el proyecto 1 y 2.

8 Referencias

- PHP.net. (s.f.). PHP: Hypertext Preprocessor. Fuente: <https://www.php.net/>
- Laravel.com. (s.f.). Laravel - The PHP Framework For Web Artisans. Fuente: <https://laravel.com/>
- Vuejs.org. (s.f.). Vue.js. Fuente: <https://vuejs.org/>
- PrimeVue. (s.f.). PrimeVue. Fuente: <https://primefaces.org/primevue/showcase/#/>
- Bootstrap. (s.f.). Bootstrap. Fuente: <https://getbootstrap.com/>
- MySQL. (s.f.). MySQL Documentation. Fuente: <https://dev.mysql.com/doc/>
- XAMPP. (s.f.). XAMPP Installers and Downloads. Fuente: <https://www.apachefriends.org/index.html>
- Laracasts.com. (s.f.). Laravel From Scratch. Fuente: <https://laracasts.com/series/laravel-8-from-scratch>
- W3Schools. (s.f.). CSS Tutorial. Fuente: <https://www.w3schools.com/css/>
- MDN Web Docs. (s.f.). CSS. Fuente: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- MDN Web Docs. (s.f.). JavaScript. Fuente: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- TutorialsPoint. (s.f.). Laravel - Quick Guide. Fuente: <https://www.tutorialspoint.com/laravel/index.htm>
- Vue Mastery. (s.f.). The Ultimate Learning Resource for Vue.js. Fuente: <https://www.vuemastery.com/>
- GitHub. (s.f.). GitHub: Where the world builds software. Fuente: <https://github.com/>
- Laravel Documentation. (s.f.). Fuente: <https://laravel.com/docs/10.x>
- Vue.js Documentation. (s.f.). Fuente: <https://vuejs.org/v2/guide/>
- PrimeVue Documentation. (s.f.). Fuente: <https://primefaces.org/primevue/showcase/#/>
- Bootstrap Documentation. (s.f.). Fuente: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

9 Guia de evaluaci3n

M3dul M05 - Entorns de Desenvolupament

UF1 - Desenvolupament de Programari

RA1 - Gestio de Projectes Web:

- Fases de desenvolupament d'una aplicaci3n informàtica.
- Avaluaci3n de la funcionalitat.
- Eines que intervenen en el desenvolupament.
- Organitzaci3n del grup de treball.

RA2 - Entorn de Desenvolupament:

- Instal·la, configura, personalitza i automatitza entorns de desenvolupament, afegeix i elimina mòduls, gestiona actualitzacions i identifica característiques d'entorns propietaris i lliures.

UF2 - Optimitzaci3n del programari

- RA1 - Verificaci3n de codi Test Unitaris
- Documenta les incidències detectades.
- Implementa proves automàtiques.
- Utilitza eines de depuraci3n.

RA2 - Optimitzaci3n de codi

- Identifica els patrons.
- Revisa el codi font utilitzant un analitzador de codi i les seves possibilitats.
- Realitza el control de versions integrat en l'entorn de desenvolupament.
- Utilitza eines per la documentaci3n del projecte.

UF3 - Introducci3n al disseny orientat a objectes

RA1 - Gestio de Projectes Web:

- Identifica, instal·la/o utilitza eines per crear, interpretar i generar codi i diagrames de classes en programaci3n orientada a objectes.

Mòdul M06 - Desenvolupament web en entorn client

UF2 - Estructures definides pel programador. Objectes

RA1 - Verificació de codi Test Unitaris

- Classifica i utilitza funcions predefinides avançades, crea i gestiona funcions i arrays, defineix i utilitza objectes amb mètodes i propietats en llenguatge orientat a objectes, depura i documenta el codi.

UF4 - Comunicació asíncrona client-servidor

RA1 - AJAX

- Analitza i utilitza mecanismes de comunicació asíncrona, gestiona objectes relacionats i les seves propietats, implementa comunicació asíncrona en aplicacions web, utilitza diferents formats per a l'enviament i recepció d'informació, i desenvolupa i depura aplicacions asíncrones.

Mòdul M07 - Desenvolupament web d'entorn servidor

UF3 - Accés a dades

RA1 - Integració BBDD:

- Analitza i utilitza tecnologies (Eloquent) per accedir a magatzems de dades, desenvolupa aplicacions amb connexions a bases de dades, recupera, filtra i publica informació web, gestiona conjunts de dades, permet actualitzacions i eliminacions d'informació, utilitza transaccions per mantenir consistència, i prova i documenta les aplicacions. Crea relacions N:M amb i sense informació extra.

UF4 - Serveis web. Pàgines dinàmiques interactives. Webs Híbrides

RA1 - API:

- Desenvolupa i verifica serveis web, coneixent les seves característiques, avantatges, tecnologies i protocols, i creant la documentació corresponent.

RA3 - Frameworks:

- Desenvolupa aplicacions web per recuperar i processar informació, crea dipòsits d'informació, utilitza llibreries de codi, programa serveis i aplicacions basats en informació de tercers, i prova, depura i documenta les aplicacions desenvolupades.

Mòdul M08 - Control de versions i documentació

UF4 - Control de versions i documentació

RA1 - GIT:

- Documenta components amb Github.
- Utilitza format Markdown i eines col·laboratives.
- Configura i gestiona un sistema de control de versions assegurant accessibilitat i seguretat.
- Documenta la seva instal·lació, configuració i ús.

Mòdul M09 - Disseny d'interfícies web

UF1 - Disseny de la interfície i estils

RA1 - Prototip:

- Crea un prototip d'una pàgina per veure quin serà l'estil visual de la pàgina web o aplicatiu web.

RA2 - HTML CSS:

- Desenvolupa tot el codi HTML i CSS per tal d'obtenir una web adaptativa tant per a smartphones com ordinadors d'escriptori.

UF2 - Elements multimèdia: creació i integració

RA1 - Arxius Multimèdia:

- Crea tots els arxius multimèdia necessaris (incloent-hi el logo en SVG) de la forma més òptima possible per tal d'obtenir el millor resultat visual amb un pes de l'arxiu del mínim possible.

UF3 - Accessibilitat i usabilitat

RA1 - Accessibilitat:

- Fes que la pàgina web sigui el més accessible possible per a smartphones i ordinadors d'escriptori.

RA2 - Usabilitat:

- Fes que la pàgina web tingui una bona usabilitat per a realitzar les funcions principals i la navegabilitat en smartphones i ordinadors d'escriptori.