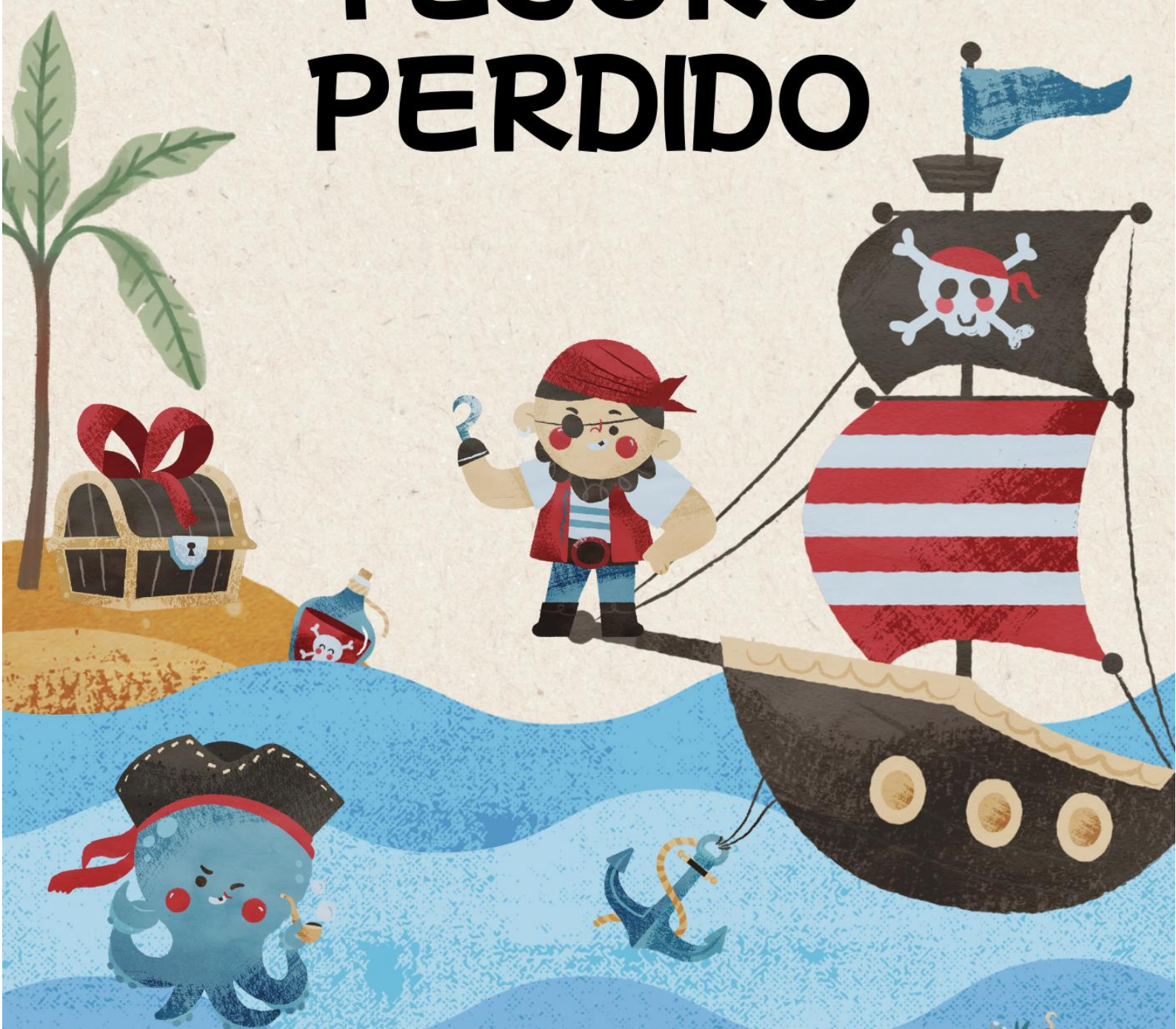


Memoria del proyecto

2ºdaw

# TESORO PERDIDO



**Manuel Gómez Navarro, Gonzalo  
Pozo Sánchez, Daniel Simón del Olmo**

## INTRODUCCIÓN

La idea de este proyecto surge de la necesidad de ofrecer a las familias una experiencia educativa innovadora y accesible, que combine la emoción de una temática atractiva como los piratas con la eficacia y los valores de la metodología Montessori. Además, la inmersión en un entorno bilingüe no solo fomenta el aprendizaje de un segundo idioma, sino que prepara a los niños para un mundo globalizado, potenciando sus habilidades de comunicación desde una edad temprana. Este campamento no es solo diversión, es también una oportunidad para aprender y crecer de manera integral.

El principal objetivo del proyecto es desarrollar una página web moderna e intuitiva que permita a las familias:

- **Informarse** fácilmente sobre las actividades, horarios y valores del campamento.
- **Inscribir a sus hijos** de manera cómoda y rápida.
- **Gestionar pagos y reservas** de forma segura.

Además, el proyecto está diseñado para responder a las necesidades del personal del campamento. Los monitores podrán acceder a información clave, consultar el cronograma y modificar actividades según las necesidades del día, lo que garantiza una mejor organización y un entorno más dinámico para los niños.

El resultado final busca no solo ser una herramienta funcional, sino también transmitir la esencia del campamento: un lugar donde la diversión y el aprendizaje se combinan en un ambiente bilingüe, inclusivo y lleno de aventuras.

## PALABRAS CLAVE

### **Backend**

Servidor web, API REST, autenticación, validación de datos, gestión de sesiones, Laravel, gestión de errores, XAMPP, PHP.

### **Frontend**

HTML5, CSS3, React, JavaScript, responsive design, UI/UX (Interfaz de usuario/Experiencia de usuario), Single Page Application (SPA), componentes reutilizables, integración con API, gestión de estado (Redux), animaciones CSS/JS, preprocesadores (SASS, LESS), accesibilidad, optimización de rendimiento (lazy loading, minificación).

### **Base de Datos (BBDD)**

MongoDB, modelado de datos, NoSQL, consultas SQL, claves primarias y foráneas, normalización, optimización de consultas.

### **Integración y despliegue**

Control de versiones (Git, GitHub), Dockerización, automatización de pruebas, entornos de desarrollo/staging/producción, Hosting (DigitalOcean), monitorización, gestión de dominios y DNS.



## Contenido

|   |    |
|---|----|
| MÓDULOS FORMATIVOS APLICADOS EN EL TRABAJO..... | 5  |
| HERRAMIENTAS Y LENGUAJES .....                  | 6  |
| ENLACE A GITHUB .....                           | 11 |

## MÓDULOS FORMATIVOS APLICADOS EN EL TRABAJO

Para la realización de este proyecto, hemos aplicado los conocimientos adquiridos en los módulos formativos de los dos años del ciclo formativo:

De las asignaturas de **primero**, hemos empleado:

- **Lenguaje de marcas:** Utilizando HTML y CSS para estructurar y diseñar las páginas del sitio web.
- **Bases de datos:** Creando y gestionando una base de datos que almacena toda la información de los usuarios, incluyendo padres, niños y monitores.
- **Entorno de desarrollo:** Aplicando Git para el control de versiones y trabajo colaborativo entre nosotros, adoptando una metodología ágil para optimizar la comunicación y el flujo de trabajo en equipo.

En cuanto a las asignaturas de **segundo**, hemos utilizado los conocimientos adquiridos en todas ellas:

- **Diseño de interfaces:** Elaborando mockups y definiendo la estética de la página web, así como los estilos y la experiencia de usuario.
- **Entorno cliente:** Utilizando JavaScript para añadir interactividad, validar formularios y mejorar la experiencia del usuario en el navegador.
- **Entorno servidor:** Aplicando PHP para el desarrollo de la lógica del servidor, incluyendo la gestión de datos, autenticación y procesos backend.
- **Despliegue de aplicaciones web:** Implementando los procesos necesarios para poner en marcha el proyecto en un entorno de producción, configurando servidores y asegurando un correcto funcionamiento en línea.

## HERRAMIENTAS Y LENGUAJES

### HERRAMIENTAS:

En este proyecto, emplearemos una combinación de herramientas y lenguajes que nos permitirán cubrir todas las fases del desarrollo, desde el diseño inicial hasta el despliegue final. Las herramientas utilizadas en el proyecto son:

**Figma:** Esta herramienta ha sido clave en la fase de diseño y planificación. Con Figma hemos elaborado:

- **La guía de estilos**, definiendo colores, tipografías, iconografía y otros elementos visuales que aseguran la consistencia en todo el proyecto.
- **User personas**, que nos permitieron identificar a nuestros usuarios clave (padres, monitores y administradores) y comprender sus necesidades y expectativas.
- **User story mapping**, ayudándonos a estructurar y priorizar las funcionalidades más importantes del proyecto desde la perspectiva del usuario.
- **El diseño de la web en general**, creando prototipos interactivos para visualizar cómo será la experiencia del usuario antes de pasar al desarrollo. Este paso nos permitió identificar posibles mejoras en la navegación y la usabilidad del sitio.

**Adobe Illustrator:** Utilizamos esta herramienta para dar color y estilo a los diseños y dibujos de la página web, asegurando una estética atractiva y personalizada que refuerza la temática pirata del campamento. Illustrator nos permitió crear ilustraciones únicas y detalladas que enriquecen la experiencia visual del **usuario**.

**XAMPP.** Es un entorno de desarrollo que hemos utilizado para simular un servidor local. Gracias a XAMPP, hemos podido trabajar con PHP y MySQL de forma eficiente, facilitando la creación, prueba y depuración de funcionalidades del backend y la conexión con la base de datos. Este entorno ha sido esencial para garantizar que las funcionalidades trabajen correctamente antes de su despliegue final.

**Visual Studio Code:** Es nuestro editor de código principal, elegido por su flexibilidad, rendimiento y compatibilidad con múltiples lenguajes. Gracias a su entorno ligero pero potente, hemos podido trabajar de manera eficiente tanto en el backend como en el

frontend. Además, hemos utilizado una serie de extensiones que mejoran aún más nuestra productividad:

- **PHP Intelephense:** Una extensión imprescindible para el desarrollo en PHP. Proporciona autocompletado, verificación de errores en tiempo real, navegación entre clases y funciones, y muchas otras características que hacen más ágil y preciso el trabajo con PHP.
- **JavaScript (ES6) Code Snippets:** Nos permite utilizar fragmentos de código predefinidos para JavaScript moderno (ES6), como funciones flecha, módulos y estructuras comunes. Esto no solo ahorra tiempo, sino que también fomenta la escritura de código más limpio y consistente.
- **Error Lens:** Una herramienta que resalta directamente en el editor los errores y advertencias en el código, mostrando mensajes detallados sin necesidad de abrir la consola. Esto nos ha permitido identificar y solucionar problemas rápidamente, reduciendo significativamente el tiempo de depuración.

**Docker:** Docker es una herramienta esencial que hemos utilizado para crear entornos de desarrollo y despliegue consistentes y portátiles. Nos permite ejecutar aplicaciones en contenedores que contienen todo lo necesario para su funcionamiento, como el sistema operativo, las dependencias y el código. Esto garantiza que el proyecto funcione de la misma manera en cualquier máquina, evitando problemas de compatibilidad.

En nuestro proyecto, Docker ha sido utilizado para:

- **Configurar entornos de desarrollo:** Creamos contenedores con servicios como PHP, MySQL y MongoDB, evitando la necesidad de configurarlos directamente en nuestras máquinas locales.
- **Simplificar el despliegue:** Hemos preparado imágenes de Docker para que el proyecto pueda ser implementado en cualquier servidor con facilidad.
- **Facilitar la colaboración:** Gracias a Docker, todo el equipo ha podido trabajar con la misma configuración, eliminando diferencias entre entornos locales.

**Draw.io:** Esta herramienta nos ha permitido diseñar y visualizar de manera clara y estructurada los diagramas relacionados con el proyecto. En particular, hemos utilizado Draw.io para modelar la base de datos y planificar otras arquitecturas clave.

En este proyecto, hemos aplicado Draw.io para:

- **El diseño de la base de datos:** Creando diagramas entidad-relación (ERD) para definir las tablas, relaciones y campos necesarios para gestionar la información de los usuarios, actividades y monitores del campamento. Esto nos ayudó a establecer una estructura sólida antes de implementar la base de datos en MongoDB.

**Discord:** Como herramienta de comunicación, usada para realizar llamadas y coordinar tareas con los compañeros del equipo, asegurando una colaboración fluida en todas las fases del proyecto.

**MongoDB:** Es una base de datos no relacional (NoSQL) que hemos elegido para gestionar la información del proyecto debido a su flexibilidad, escalabilidad y facilidad de uso. MongoDB almacena los datos en formato de documentos JSON, lo que permite manejar estructuras complejas de manera más natural y eficiente en comparación con bases de datos relacionales.

En el contexto de nuestro proyecto, hemos utilizado MongoDB para:

- **Gestión de usuarios:** Almacenar la información de los padres, monitores y administradores del campamento, incluyendo datos personales y credenciales de acceso.
- **Registro de actividades:** Crear una colección para las actividades del campamento, con detalles como horarios, monitores asignados y descripciones.
- **Flexibilidad en la estructura de datos:** MongoDB nos permitió trabajar con datos que no siempre siguen una estructura fija, facilitando la inclusión de nuevas funcionalidades sin requerir modificaciones complicadas en el esquema.
- **Rápido acceso a la información:** Gracias a sus índices y consultas eficientes, hemos podido garantizar que los datos se recuperen rápidamente, incluso con grandes volúmenes de información.



## LENGUAJES:

**PHP:** es el lenguaje de programación del lado del servidor que hemos utilizado para construir la lógica del backend del proyecto. Su integración sencilla con bases de datos y su capacidad para generar contenido dinámico lo hacen ideal para desarrollar aplicaciones web como la nuestra.

En este proyecto, PHP ha sido clave para:

- **Gestión de usuarios:** Procesar el registro, autenticación y actualización de datos de padres, monitores y administradores del campamento.
- **Conexión con bases de datos:** Implementar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en MongoDB, facilitando la interacción con las colecciones de usuarios, actividades y más.
- **Procesamiento de formularios:** Validar y gestionar los datos enviados por los usuarios, como las inscripciones de los niños al campamento, para garantizar su integridad y seguridad.
- **Generación de contenido dinámico:** Renderizar las páginas con datos específicos para cada usuario, como el cronograma de actividades personalizado o la información del perfil.
- **Integración con otras tecnologías:** PHP se complementa con Docker y herramientas como XAMPP para probar y desplegar las funcionalidades del servidor de forma local antes de implementarlas.

**JavaScript:** Es el lenguaje de programación del lado del cliente que hemos utilizado para agregar interactividad y dinamismo a las páginas web del proyecto. Gracias a JavaScript, hemos podido mejorar la experiencia de usuario haciendo que las funcionalidades sean rápidas, responsivas y atractivas visualmente.

En el proyecto del campamento, JavaScript ha sido clave para:

- **Validación de formularios:** Verificar en tiempo real los datos introducidos por los usuarios (como inscripciones o información de contacto), asegurándonos de que sean correctos antes de enviarlos al servidor.

- **Interactividad dinámica:** Implementar funcionalidades como la selección de actividades, la actualización en vivo de cronogramas o la visualización de mensajes personalizados para los usuarios.
- **Control de la interfaz:** Gestionar el comportamiento de elementos visuales, como mostrar u ocultar secciones de la página, activar animaciones o cambiar estilos según las interacciones del usuario.
- **Comunicación con el servidor:** Usar **fetch** o **AJAX** para enviar y recibir datos de forma asíncrona, permitiendo que ciertas funcionalidades (como la matriculación o consulta de actividades) se ejecuten sin necesidad de recargar la página.
- **Gestión de eventos:** Detectar y manejar eventos como clics, desplazamientos o entradas de texto, proporcionando una experiencia de usuario más intuitiva y fluida.

Los **frameworks** utilizados en nuestro proyecto son:

- **React** es una biblioteca de JavaScript diseñada para construir interfaces de usuario interactivas y dinámicas. Hemos elegido React para el desarrollo del frontend del proyecto debido a su enfoque en componentes reutilizables, su eficiencia y su capacidad para manejar aplicaciones web complejas.
- **Laravel:** Laravel es un framework de PHP que hemos utilizado para desarrollar el backend del proyecto. Es conocido por su sintaxis elegante y su robusta estructura, lo que facilita la creación de aplicaciones web de manera rápida y segura. Laravel nos ha permitido desarrollar funcionalidades complejas sin tener que reinventar la rueda, gracias a su enfoque en el desarrollo rápido y las mejores prácticas.

Esta combinación de herramientas y tecnologías ha sido seleccionada para aprovechar lo mejor de cada una y garantizar un desarrollo eficiente, colaborativo y de alta calidad.

ENLACE A GITHUB

<https://github.com/manuelgomeznararro/CampamentoPirata/tree/dev>