



**Universidad  
Europea** MADRID

## **PROYECTO INTEGRADOR**

**2 DAW**

**[WOOKY DOOKY]**



### **Integrantes del Proyecto:**

IAN

FANGYUE WU

JING XIANG WANG

### **Profesores:**

Sara Villanueva

Irene del Rincón

Raquel Cerda Losa

## Introducción

Este documento presenta el desarrollo y análisis de Wooky Dooky, una página web diseñada para servir como herramienta de apoyo integral en el contexto de un campamento. El proyecto tiene como finalidad proporcionar servicios clave, compartir contenido educativo y ofrecer una plataforma interactiva que permita una comunicación eficiente y una experiencia enriquecedora para padres, monitores y participantes. La página busca replicar las funcionalidades de un aula virtual, adaptándolas a las necesidades específicas del entorno campestre y recreativo.

El desarrollo del diseño visual y funcional de la página web se llevó a cabo utilizando Figma, una herramienta que facilitó la creación del mockup y de una guía de estilos. Estas herramientas sirvieron como base para conceptualizar el aspecto final del proyecto, permitiendo una planificación precisa y una alineación estética coherente con los objetivos del campamento.

El propósito principal de esta memoria es documentar cada etapa del proceso de creación de la página web, desde la concepción inicial hasta la implementación final, incluyendo un análisis detallado de los desafíos enfrentados y las soluciones aplicadas. Además, se busca extraer aprendizajes significativos que puedan ser útiles para futuros proyectos, especialmente aquellos relacionados con la digitalización de experiencias educativas y recreativa.

El proyecto se organizó en cuatro sprints, estructurados bajo una metodología ágil que permitió avanzar de manera ordenada y eficiente. Cada sprint abordó diferentes fases del desarrollo, como la planificación inicial, el diseño visual, la implementación técnica y la revisión final, asegurando así una evolución progresiva y bien fundamentada del producto.

En definitiva, este documento no solo detalla el proceso técnico y creativo detrás de Wooky Dooky, sino que también ofrece una reflexión crítica sobre las estrategias utilizadas, con el objetivo de enriquecer futuros desarrollos que busquen combinar tecnología y experiencias presenciales de manera efectiva.

**Palabras clave:**

Sprint, metodologías, HTML, CSS, JavaScript, PHP, XAMP, Figma, MySQL, diagramas, bases de datos, programación, diseño.

# Índice

## Tabla de contenido

Introducción .....	1
Palabras clave: .....	2
Módulos formativos incluidos .....	4
Herramientas y Lenguajes Utilizados en el Proyecto .....	6
Componentes del equipo y aportación realizada por cada estudiante .....	8
Fases del proyecto .....	10
Diseño de interfaces .....	14
Modelo de datos .....	15
Diagrama de UML .....	16
Diagrama de Entidad-Relación .....	18
Wireframes y estructura visual: .....	20
Guía de estilos y coherencia visual: .....	20
Planificación del Desarrollo .....	25
Introducción al desarrollo .....	28
Conclusiones .....	33
Bibliografía .....	35
Anexos: .....	37
Pruebas manuales del PHP .....	37
Instalación de apache .....	46
Manuar de usuario .....	48
Manual para Padres - Guía de Uso y Funcionalidades .....	49
Manual para Monitores - Guía de Uso y Funcionalidades .....	54
Manual para Administradores - Guía de Uso y Funcionalidades .....	56

## **Módulos formativos incluidos**

### **Diseño de interfaces web:**

En esta asignatura se abordó el diseño de interfaces centradas en el usuario (UI), utilizando herramientas como Figma para crear prototipos interactivos y mockups. Se aprendieron principios de diseño como la coherencia visual, la jerarquía de información y la experiencia del usuario (UX). También se realizó un estudio de mercado para analizar las necesidades de los usuarios, permitiendo ajustar el diseño a sus expectativas. A través de ejercicios prácticos, se adquirieron competencias en la planificación estética y funcional de aplicaciones y páginas web, creando diseños intuitivos, accesibles y adaptados al propósito del proyecto.

### **Entorno Cliente:**

En esta asignatura se profundizó en JavaScript, el lenguaje de programación utilizado para añadir interactividad a las páginas web. Se exploraron temas como el manejo del DOM (Document Object Model), la manipulación de eventos y la creación de animaciones. También se abordaron tecnologías relacionadas como AJAX, permitiendo la carga dinámica de contenido sin recargar la página. Los conocimientos adquiridos permitieron implementar funcionalidades avanzadas, como formularios dinámicos, validaciones en tiempo real y mejoras en la experiencia del usuario mediante interfaces más responsivas e interactivas.

### **Desarrollo de Entorno Servidor:**

En esta asignatura se exploró el lenguaje PHP (Hypertext Preprocessor), ampliamente utilizado para la programación del lado del servidor. Se adquirieron conocimientos para gestionar la lógica detrás de las páginas web dinámicas, como el procesamiento de formularios, la validación de datos y la interacción con bases de datos. Además, se trabajó en la implementación de sesiones y autenticación de usuarios, aspectos fundamentales para la seguridad de las aplicaciones. A través de proyectos prácticos, se profundizó en la integración de PHP con otras tecnologías como HTML, logrando páginas web interactivas y personalizadas.

### **Entorno Desarrollo:**

En el desarrollo del proyecto se aplicaron metodologías ágiles, específicamente Scrum, dividiendo el trabajo en sprints para fomentar la colaboración y mejorar continuamente. Se siguieron principios de Clean Code para escribir código claro, legible y mantenible, complementados con diagramas UML y de flujo que facilitaron la planificación y comunicación de la estructura

del sistema. Git se utilizó como herramienta de control de versiones, permitiendo gestionar cambios mediante ramas, repositorios remotos y revisiones colaborativas. Además, se llevaron a cabo pruebas unitarias, de integración y de aceptación para garantizar la funcionalidad, calidad y cumplimiento de los requisitos del proyecto.

**Marcas de lenguaje:**

Esta asignatura se centra en el aprendizaje y aplicación de los lenguajes base para el desarrollo web. HTML (HyperText Markup Language) se utiliza para estructurar el contenido de las páginas, definiendo elementos como encabezados, párrafos, tablas y formularios. CSS (Cascading Style Sheets), por su parte, permite estilizar la estructura creada, definiendo colores, tipografías, tamaños, márgenes, y otros aspectos visuales. Durante esta formación, se adquirieron habilidades para combinar ambas tecnologías, logrando sitios web visualmente atractivos y funcionales. Además, se trabajaron conceptos de diseño responsivo, asegurando que las páginas se adapten a diferentes dispositivos y tamaños de pantalla.

**Base de Datos:**

Esta materia se centró en el aprendizaje del sistema de gestión de bases de datos MySQL. Se desarrollaron habilidades para crear, estructurar y administrar bases de datos, esenciales para el almacenamiento y recuperación de información en aplicaciones web. Se estudiaron conceptos clave como tablas, relaciones, consultas SQL, claves primarias y foráneas, así como procedimientos almacenados. Además, se trabajó en la conexión entre MySQL y lenguajes como PHP, lo que permitió integrar datos dinámicos en el proyecto, gestionando de forma eficiente la información de los usuarios y el contenido.

## Herramientas y Lenguajes Utilizados en el Proyecto

Para garantizar un desarrollo eficiente, escalable y seguro, se empleó un conjunto de tecnologías cuidadosamente seleccionadas. Estas herramientas abarcan desde lenguajes de programación hasta frameworks, bases de datos, entornos de desarrollo y metodologías que facilitaron la gestión del proyecto.

### 1. Lenguajes de Programación

#### Frontend: Creación de la Interfaz de Usuario

El frontend del proyecto fue desarrollado utilizando un conjunto de tecnologías que garantizan una experiencia de usuario fluida y una interfaz atractiva:

- **HTML:** Lenguaje fundamental para la estructuración de los contenidos en la web. Su uso permitió definir la jerarquía de elementos en cada página del sistema.
- **CSS:** Utilizado para el diseño y la presentación visual de la plataforma. Se aplicaron técnicas avanzadas de diseño responsivo para optimizar la visualización en distintos dispositivos.
- **JavaScript:** Lenguaje de programación dinámico empleado para añadir interactividad al sistema. Se utilizaron métodos modernos, como `fetch()` para la comunicación con el backend.

#### Backend: Lógica y Procesamiento de Datos

Para la parte del servidor y la gestión de la lógica del negocio, se emplearon las siguientes tecnologías:

- **PHP:** Lenguaje de programación ampliamente utilizado en el desarrollo web. Se encargó del procesamiento de datos en el servidor, la validación de formularios y la gestión de sesiones de usuario.
- **MySQL:** Lenguaje utilizado para la gestión y manipulación de la base de datos. Se diseñó un esquema relacional optimizado para almacenar información estructurada de usuarios, actividades y registros históricos.

### 2. Arquitectura del Proyecto

El sistema se organizó bajo una estructura basada en el **modelo Vista-Controlador**, lo que permitió una separación clara entre la lógica del negocio, la presentación y el control del flujo de datos. Esto facilitó la escalabilidad y el mantenimiento del código.

Dentro del proyecto, los archivos se distribuyeron en diferentes carpetas dentro de la carpeta principal denominada **CODE**, la cual contiene:

- **HTML, CSS y JavaScript:** Archivos encargados de la interfaz y experiencia de usuario.
- **PHP:** Lógica del backend y conexión con la base de datos.
- **SQL:** Scripts de base de datos para la creación y gestión de tablas.
- **Assets:** Carpeta donde se almacenan imágenes, fuentes y otros recursos necesarios para el diseño y funcionamiento del sistema.

### 3. Herramientas y Frameworks Utilizados

Para optimizar el desarrollo y la implementación del proyecto, se hizo uso de diversas herramientas y frameworks que facilitaron la programación, la gestión de datos y la experiencia del usuario.

#### Herramientas de Desarrollo

- **Figma:** Utilizado en la fase de diseño para la creación de wireframes interactivos, lo que permitió visualizar la estructura de la interfaz de usuario antes de su implementación.
- **Visual Studio Code:** Entorno de desarrollo integrado (IDE) utilizado para la programación. Ofreció una experiencia de desarrollo optimizada gracias a sus extensiones para PHP, JavaScript y CSS.
- **Git y GitHub:** Herramientas esenciales para el control de versiones y la colaboración en equipo. Git permitió gestionar cambios en el código de manera eficiente, mientras que GitHub facilitó la integración de nuevas funcionalidades sin afectar la estabilidad del sistema.

#### Bases de Datos y Seguridad

- **MySQL:** Base de datos relacional utilizada para almacenar y gestionar la información del sistema. Se implementaron técnicas de indexación y consultas optimizadas para mejorar el rendimiento.
- **Encriptación de Datos:** Se utilizaron métodos como el hashing de contraseñas para proteger la información sensible de los usuarios.

#### Infraestructura y Despliegue

- **Servidor Web Apache:** Utilizado para alojar y ejecutar el sistema en un entorno estable y seguro.



## **Componentes del equipo y aportación realizada por cada estudiante**

El equipo de trabajo está conformado por tres integrantes: Ian, Antonia y Fang Yue. A lo largo del desarrollo del proyecto, hemos colaborado estrechamente en todas las áreas, asegurándonos de que cada aspecto fuera abordado de manera integral. Si bien todos hemos trabajado en conjunto y participado en diferentes etapas del desarrollo, es innegable que cada uno ha tenido un rol más destacado en las áreas que mejor se alinean con sus habilidades y experiencia.

Desde el inicio del proyecto, nos organizamos de manera estratégica para optimizar nuestro tiempo y esfuerzo. Adoptamos una metodología de trabajo basada en la comunicación constante y en la división equitativa de tareas, lo que nos permitió avanzar de manera eficiente y coordinada. Cada uno de nosotros aportó su conocimiento y creatividad en distintas fases del desarrollo, asegurando así que el resultado final fuera sólido, funcional y bien estructurado.

A continuación, se detallan las responsabilidades y contribuciones específicas de cada integrante del equipo:

### **Ian – Especialista en Backend y Gestión de la Lógica del Proyecto**

Ian asumió el rol principal en el desarrollo del backend, siendo el encargado de diseñar, estructurar y optimizar la base de datos. Su trabajo fue fundamental para garantizar que el almacenamiento de información fuera eficiente, seguro y bien organizado. Además, se enfocó en la lógica central del proyecto, asegurándose de que los distintos módulos del sistema se comunicaran correctamente y que la funcionalidad del software fuera fluida y libre de errores.

Entre sus tareas más relevantes se encuentran:

- La creación y gestión de la base de datos, asegurando que fuera escalable y optimizada para el rendimiento.
- La implementación de los distintos servicios necesarios para la comunicación entre el servidor y el cliente.
- La optimización del código para garantizar tiempos de respuesta rápidos y una arquitectura eficiente.
- La solución de problemas y el mantenimiento del backend para asegurar su estabilidad a lo largo del desarrollo.
- Gracias a su labor, el proyecto cuenta con una base sólida sobre la cual funcionan todas las demás capas del sistema.

**Antonia – Apoyo Integral en Backend y Frontend**

Antonia desempeñó un papel clave en el equipo gracias a su versatilidad y capacidad para adaptarse a diferentes áreas del desarrollo. Su perfil polifacético le permitió contribuir tanto en el backend como en el frontend, actuando como un puente entre ambas partes y facilitando la integración de las diferentes tecnologías utilizadas en el proyecto.

Entre sus principales aportes se encuentran:

- Asistencia en la implementación de la lógica del backend, colaborando con Ian en la estructuración de los datos y el desarrollo de la web.
- Desarrollo de funciones y características clave en el frontend, asegurándose de que la interfaz fuera intuitiva, atractiva y funcional.
- Integración del backend con el frontend, asegurando que la comunicación entre ambas partes se realizara de manera eficiente y sin errores.
- Su capacidad para adaptarse a diferentes desafíos y su enfoque en la eficiencia fueron cruciales para el éxito del proyecto, ya que permitió que todas las partes del sistema estuvieran bien coordinadas y funcionaran sin problemas.

**Fang Yue – Desarrollo de Frontend y Estructuración de Componentes**

Fang Yue se especializó en la parte visual e interactiva del proyecto, encargándose del desarrollo del frontend y la estructuración de los componentes de la web. Su trabajo fue fundamental para garantizar que la interfaz de usuario fuera moderna, intuitiva y atractiva, asegurando una experiencia de usuario fluida y agradable.

Entre sus contribuciones más importantes destacan:

- La implementación de los componentes de la interfaz utilizando JavaScript y tecnologías asociadas.
- La organización y estructuración del código del frontend para mantener una arquitectura clara y modular.
- La optimización del diseño visual para que la aplicación resultara fácil de usar y agradable a la vista.
- La implementación de interacciones dinámicas que mejoran la experiencia del usuario y facilitan la navegación dentro de la aplicación.

## Fases del proyecto

### Estudio de mercado

#### 1. Definir público objetivo

En este caso, el público objetivo principal son niños de 6 a 7 años, pero también debes considerar a sus padres, quienes son los que tomarán la decisión de inscripción.

#### Características del público:

##### Niños de 6-7 años:

- Están en una etapa de desarrollo físico, emocional y social.
- Les gustan las actividades lúdicas, creativas y al aire libre.
- Necesitan supervisión constante y un entorno seguro.

##### Padres:

- Buscan seguridad, confianza y un entorno educativo o enriquecedor.
- Valorarán la calidad de las actividades, el personal y las instalaciones.
- Prestarán atención a la relación calidad-precio.

#### 2. Investiga las necesidades y preferencias

Para entender mejor qué buscan los padres y los niños, puedes hacer lo siguiente:

##### Para los niños:

- Actividades que disfrutan: juegos al aire libre, manualidades, deportes, cuentos, etc.
- Preferencias temáticas: aventura, naturaleza, arte, etc.
- Duración ideal del campamento: medio día, día completo o con pernoctación.

##### Para los padres:

- Seguridad: instalaciones adecuadas, personal cualificado, protocolos de emergencia.
- Horarios compatibles con su rutina.
- Servicios adicionales: transporte, comida, flexibilidad de horarios.
- Precio: cuánto están dispuestos a pagar por un campamento de calidad

##### Herramientas:

- Encuestas a padres de niños de 6-7 años (Google Forms, Typeform).
- Entrevistas en colegios, parques o centros de actividades infantiles.
- Observación de comportamientos en redes sociales o grupos de padres.

### 3. Analiza a la competencia

Investiga otros campamentos en Madrid dirigidos a niños de 6-7 años. Algunos aspectos clave a analizar:

#### Tipos de campamentos:


- Urbanos (en la ciudad) o en la naturaleza.
- Temáticos (deportivos, artísticos, científicos, etc.).
- Con o sin pernoctación.
- Actividades que ofrecen:
- Juegos, talleres, excursiones, etc.

#### Precios:

- Coste por día, semana o mes.
- Descuentos por hermanos o inscripciones anticipadas.

#### Herramientas:

- Búsquedas en Google ("campamentos para niños de 6 años en Madrid").
- Redes sociales de campamentos competidores.
- Plataformas como Google Reviews o foros de padres.




Facilita el número del teléfono del centro, aunque podría pasar algo desapercibido al no ser muy llamativo

El menu corta la imagen

Cuadro con más información y eventos


Fangyue Wu



Hace bien indicando en que paso se encuentra en cada momento

Debería de poner como debería de ser el formato de los datos

Fangyue Wu



Bien diseñado, mostrando de forma visual las colaboraciones

Demasiado texto, debería de existir una opción para poder obtener más información si lo deseamos

Antonio



#### 4. Identifica tendencias en campamentos infantiles

Mantente al día con las tendencias actuales en campamentos para niños pequeños. Algunas ideas:

**Sostenibilidad:** Actividades relacionadas con el cuidado del medio ambiente.

**Tecnología:** Talleres de robótica o programación adaptados a niños.

**Educación emocional:** Actividades que fomenten la inteligencia emocional y las habilidades sociales.

**Multiaventura:** Actividades al aire libre como senderismo, tirolinas o juegos de agua.

**Fuentes de información:**

- Blogs y revistas sobre educación infantil.
- Ferias y eventos de ocio y educación.
- Grupos de padres en redes sociales.

#### 5. Estrategia de precios

Investiga los precios de la competencia y define una estrategia de precios para tu campamento. Considera:

- Costes de operación (personal, materiales, instalaciones, etc.).
- Precios de la competencia (para posicionarte de manera competitiva).
- Descuentos o promociones (por inscripción anticipada, hermanos, etc.).
- Ejemplo de precios en Madrid:
  - Campamentos urbanos: €100-€200 por semana.
  - Campamentos con pernoctación: €300-€500 por semana.



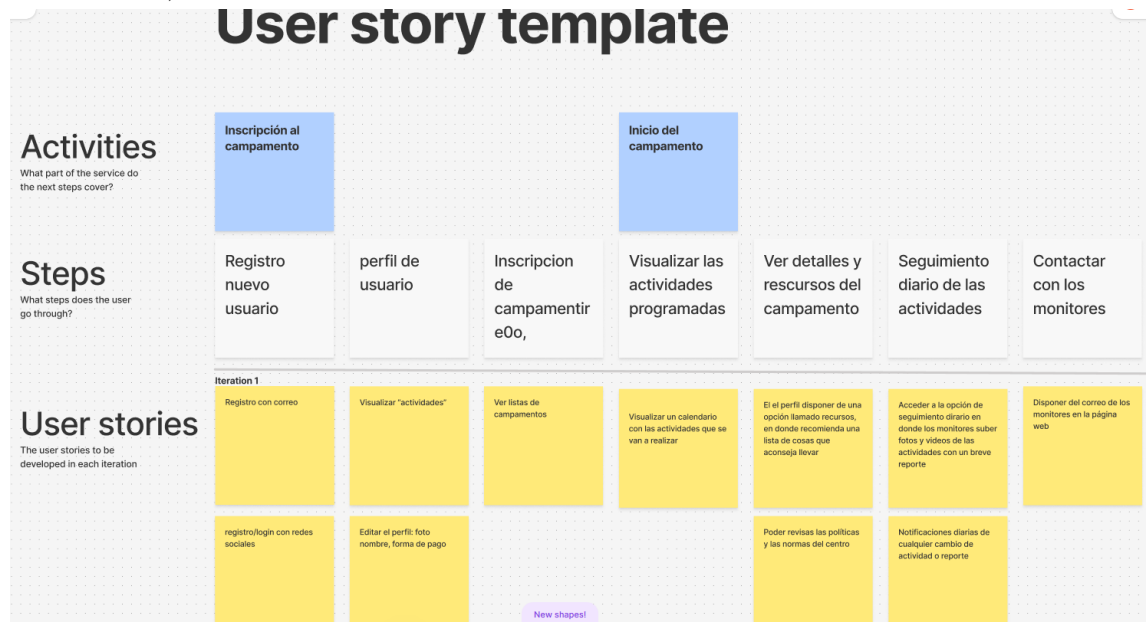
## Diseño de interfaces

El proceso de diseño se llevó a cabo siguiendo una metodología estructurada, centrada en garantizar tanto la funcionalidad como la experiencia del usuario. A continuación, se detallan las etapas clave:

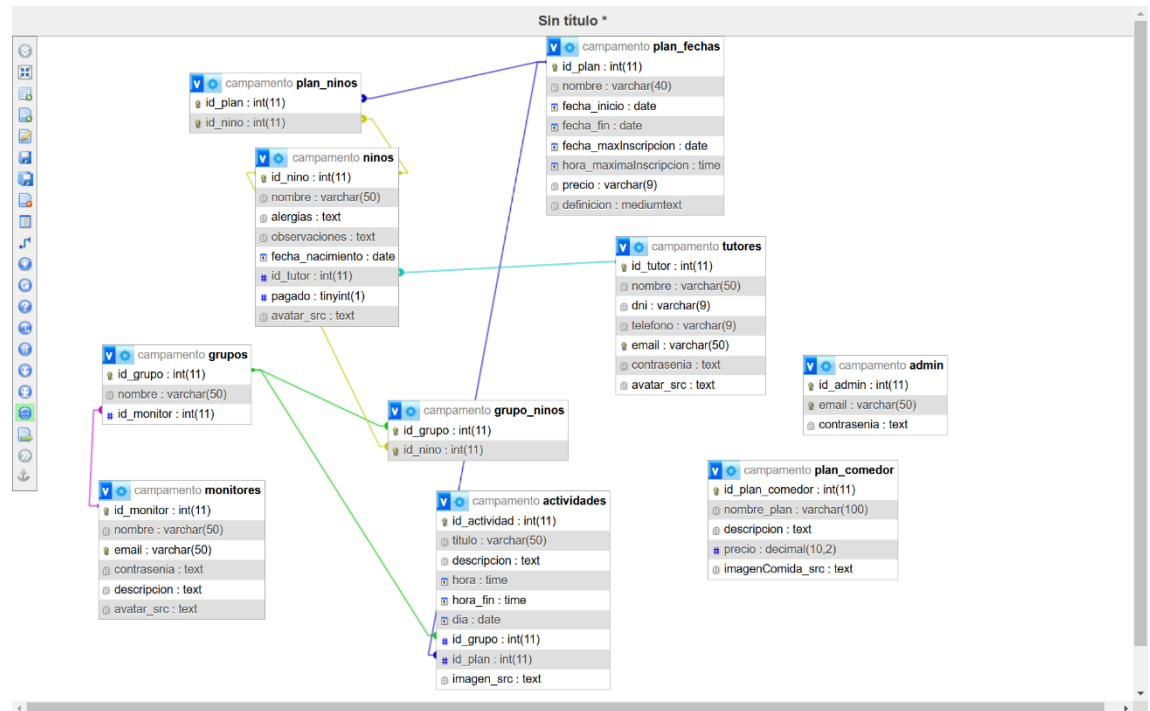
### Análisis de los requisitos:

Se inició con un estudio exhaustivo de los requisitos de los dos perfiles principales: los monitores (encargados de gestionar actividades diarias) y los administradores del centro (responsables de la configuración global y seguridad).

Mediante entrevistas y cuestionarios, se identificaron puntos críticos: acceso rápido a información, claridad en la navegación y herramientas específicas para cada rol (ej.: paneles de control diferenciados).



## Modelo de datos



Este modelo de datos gestiona un sistema de campamento. Las tablas principales y sus funciones son:

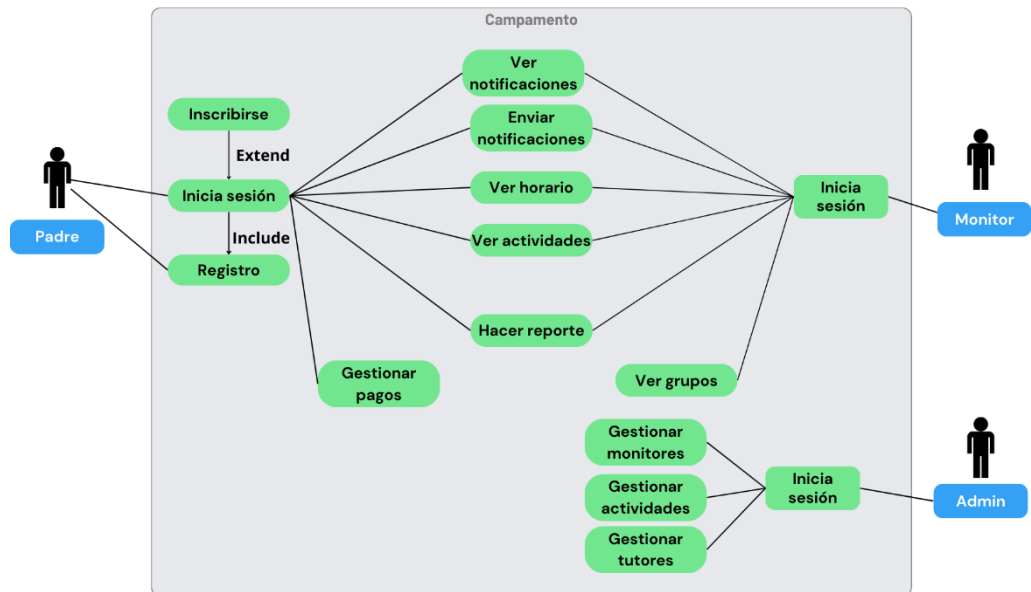
- **Administradores** (campamento\_admin): Contiene credenciales de los Administradores del sistema.
- **Monitores** (campamento\_monitores): Gestiona los datos de los monitores, incluyendo su información de contacto y credenciales.
- **Grupos** (campamento\_grupos): Define los grupos de niños y asigna un monitor responsable a cada grupo.
- **Niños** (campamento\_ninos): Almacena información personal de los niños, como alergias, observaciones, fecha de nacimiento, tutor asociado y si han pagado.
- **Tutores** (campamento\_tutores): Registra los datos de contacto de los tutores, incluyendo su DNI, teléfono y correo electrónico.
- **Actividades** (campamento\_actividades): Contiene la información sobre las actividades del campamento, incluyendo título, descripción, horarios, grupo asignado y precio.
- **Planes** (campamento\_plan\_fechas): Define los planes del campamento con fechas clave, precio y descripción general.
- **Relación Grupos-Niños** (campamento\_grupo\_ninos): Vincula a los niños con sus respectivos grupos dentro del campamento.
- **Relación Planes-Niños** (campamento\_plan\_ninos): Establece qué niños están inscritos en qué planes del campamento.
- **Plan de Comedor** (campamento\_plan\_comedor): Almacena



información sobre los planes de alimentación del campamento, incluyendo descripción, precio e imagen de referencia. **Monitores** (campamento\_monitores): Gestiona los datos de los monitores.

- **Grupos** (campamento\_grupos): Define grupos de niños, asignados a monitores.
- **Niños** (campamento\_ninos): Almacena información personal, alergias, tutor asociado y plan inscrito.
- **Tutores** (campamento\_tutores): Registra los datos de contacto de los tutores.
- **Actividades** (campamento\_actividades): Detalla actividades con su monitor y plan asociado.
- **Planes** (campamento\_plan\_fechas): Define planes con fechas importantes.
- **Relación Grupos-Niños** (campamento\_grupo\_ninos): Vincula niños con grupos.

## Diagrama de UML



### Actores:

**Padre:** Inscribe a sus hijos, gestiona pagos y consulta información.

**Monitor:** Supervisa grupos, envía notificaciones y crea reportes.

**Admin:** Gestiona monitores, actividades y tutores.

### Casos de Uso Clave:

#### Padre:

- **Inscribirse:** Registra al hijo en el campamento.
- **Ver notificaciones, horarios y actividades:** Consulta información relacionada.
- **Gestionar pagos:** Realiza y verifica pagos.

#### Monitor:

- **Hacer reporte:** Genera reportes de actividades o grupos.
- **Enviar notificaciones:** Informa a padres y admin.

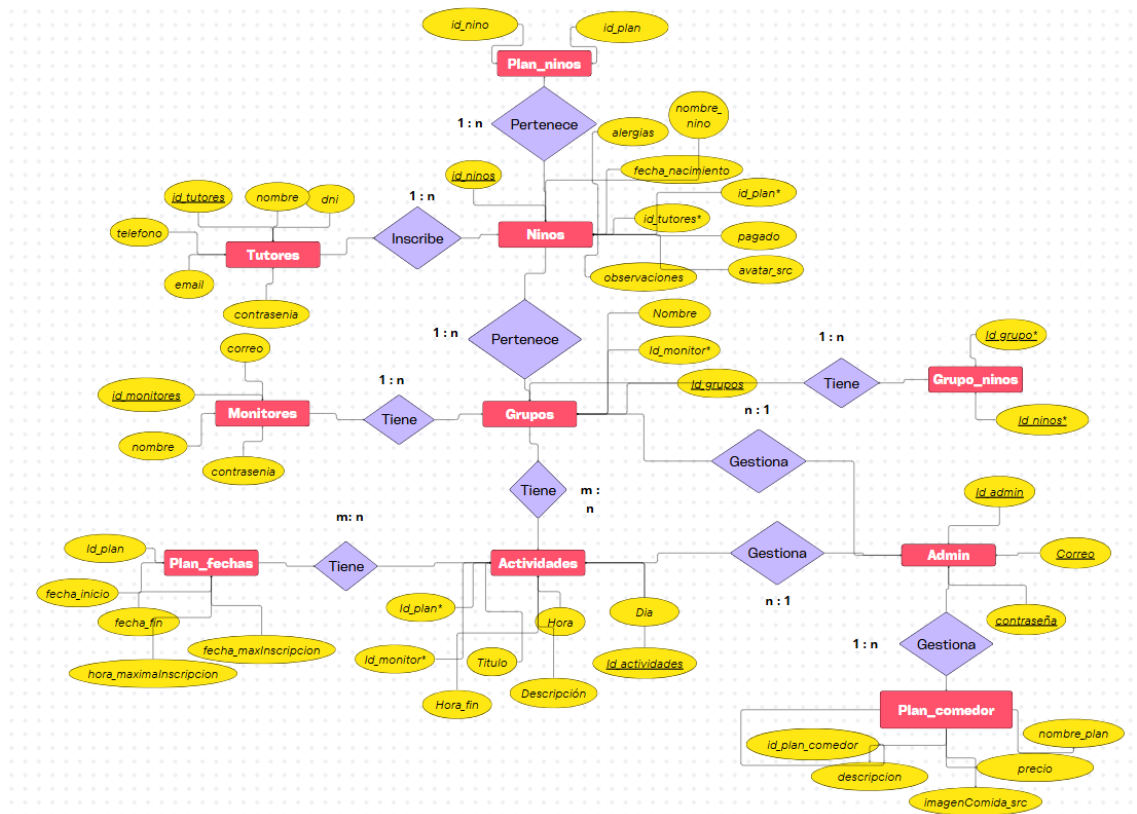
#### Admin:

**Gestionar monitores, actividades y tutores:** Administra recursos y usuarios.

### Atributos Importantes:

- **Notificaciones:** Fecha, contenido, destinatario.
- **Actividades:** Título, horario, descripción.
- **Pagos:** Estado, monto, método.
- **Usuarios:** Nombre, correo, contraseña.

## Diagrama de Entidad-Relación



### Tutores

**Tutores** (Id\_tutores, nombre, dni, telefono, email, contrasenia)

PK: Id\_tutores

### Monitores

**Monitores** (Id\_monitores, nombre, contrasenia)

PK: Id\_monitores

### Niños

**Niños** (Id\_niños, nombre\_niño, fecha\_nacimiento, Id\_tutores\*, Id\_plan\*, alergias, observaciones, pagado, avatar\_src)

PK: Id\_niños

FK: Id\_tutores\*, Id\_plan\*

### Grupos

**Grupos** (Id\_grupos, Nombre, Id\_monitor\*)

PK: Id\_grupos

FK: Id\_monitor\*

### Grupo\_Niños

**Grupo\_Niños** (Id\_grupo\*, Id\_niños\*)

PK: (Id\_grupo, Id\_niños)

FK: Id\_grupo\*, Id\_niños\*

**Actividades**

**Actividades** (Id\_actividades, Hora, Dia, Titulo, Descripcion,  
Id\_plan\*, Id\_monitor\*)

PK: Id\_actividades

FK: Id\_plan\*, Id\_monitor\*

**Plan\_fechas**

**Plan\_fechas** (Id\_plan, fecha\_inicio, fecha\_fin,  
hora\_max\_inscripcion, fecha\_max\_inscripcion)

PK: Id\_plan

**Admin**

**Admin** (Id\_admin, Correo, contrasenia)

PK: Id\_admin

**Admin**

**Admin** (Id\_admin, Correo, contrasenia)

PK: Id\_admin

**Plan\_ninos**

**Plan\_ninos** (Id\_plan, Id\_ninos )

PK: Id\_plan, Id\_ninos

**Plan\_comedor**

**Plan\_comedor** (**Id\_plan\_comedor**, nombre\_plan, descripcion,  
precio, imagenComida\_src)

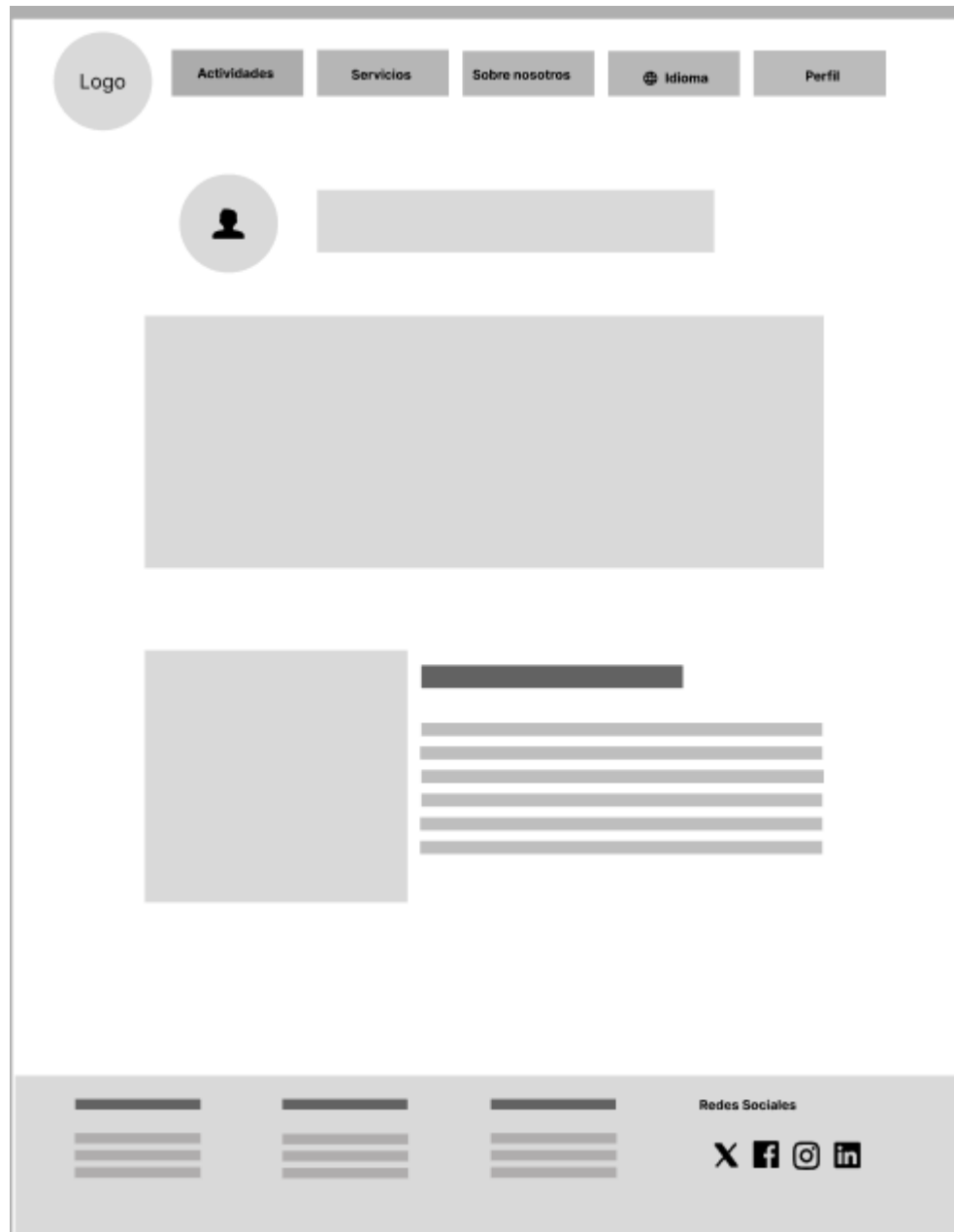
PK: **Id\_plan\_comedor**

## Wireframes y estructura visual:

Se desarrollaron bocetos preliminares (wireframes) para definir la distribución de elementos en cada página (menús, secciones prioritarias, espacios interactivos).

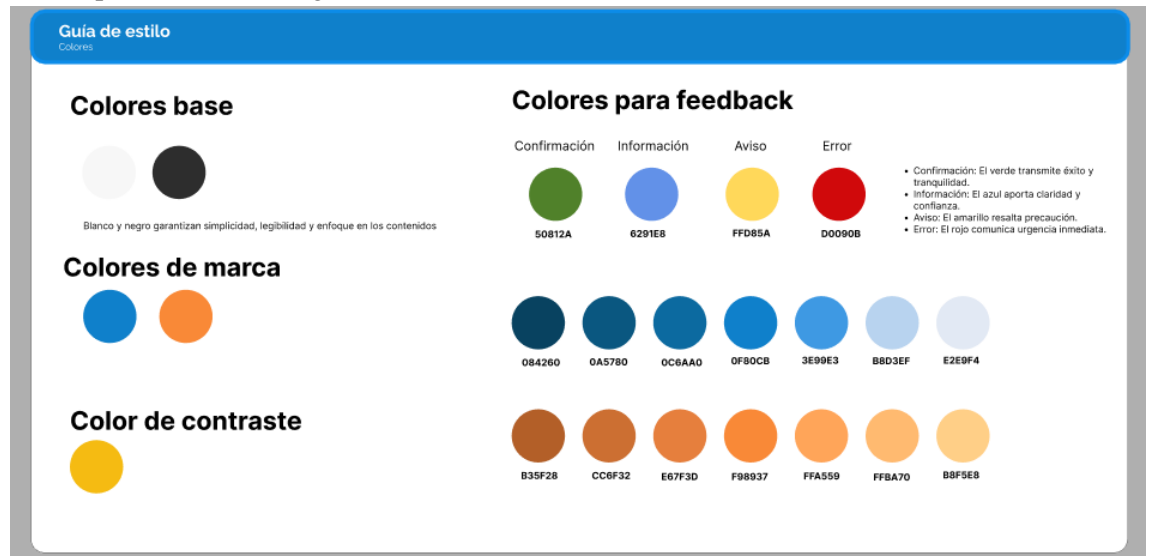
Se priorizó la usabilidad: por ejemplo, se diseñaron rutas intuitivas para tareas frecuentes (como registrar asistencia o generar reportes) y se evitó la sobrecarga de información.

Herramientas como Figma permitieron ajustar iterativamente los diseños basados en feedback inicial.

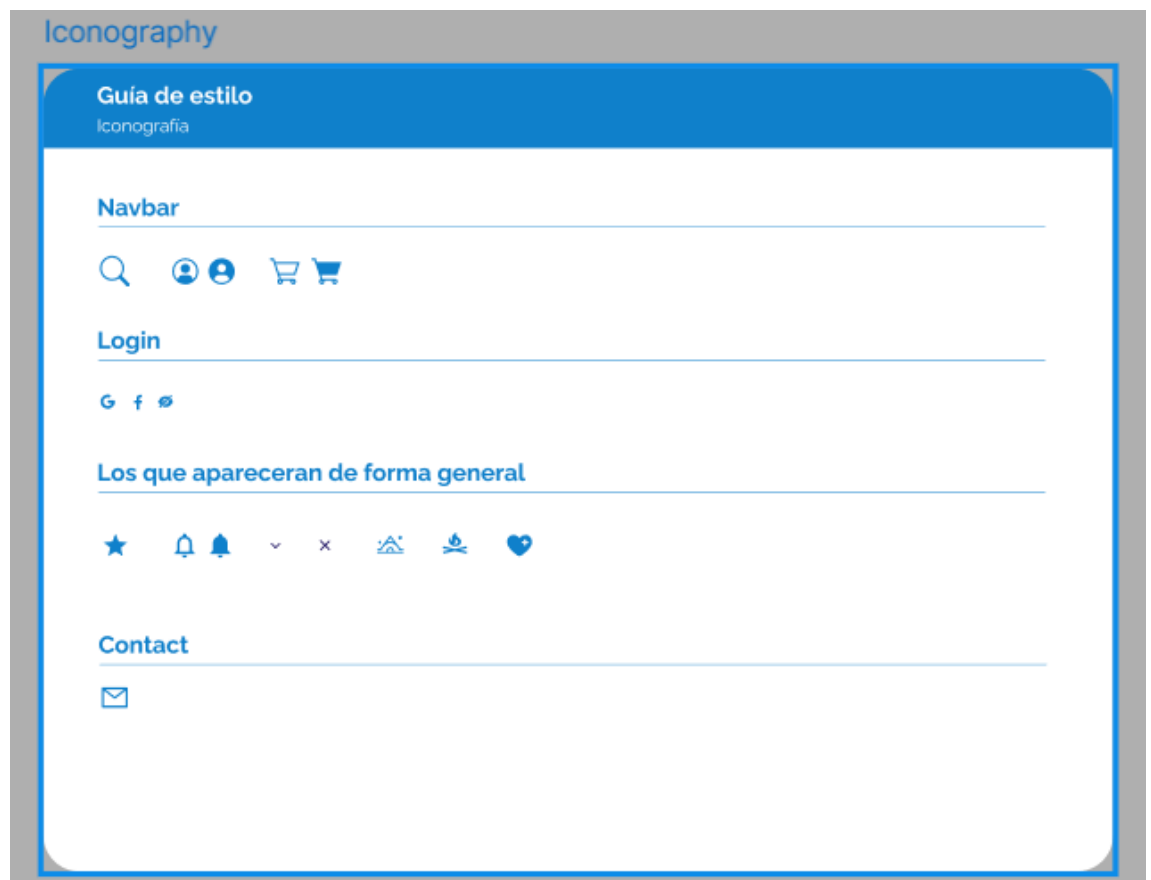


## Guía de estilos y coherencia visual:

Paleta de colores: Se seleccionaron tonos institucionales para reforzar la identidad del centro, combinados con contrastes suaves para reducir la fatiga visual



Iconografía: Se optó por iconos minimalistas y universales (como una campaña para notificaciones o un calendario para agendas) para garantizar comprensión inmediata.



### Componentes interactivos:

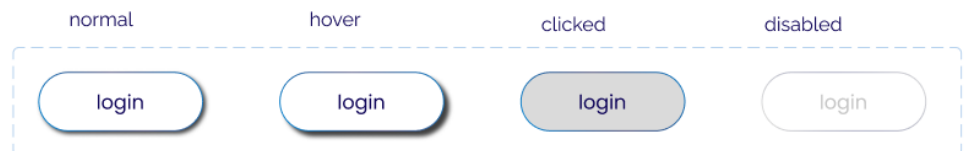
Botones: Diseñados con jerarquía clara (primarios, secundarios) y tamaños adaptables a dispositivos móviles.

Efectos hover: Microinteracciones sutiles (cambios de color o sombras) para guiar al usuario sin distraer.

#### Botones primarios



#### Botones secundarios



### **Prototipado final (mockup):**

Se integraron todos los elementos anteriores en un modelo interactivo de alta fidelidad, simulando la experiencia real de uso.


Se incluyeron detalles como:

**Responsividad:** Adaptación fluida a pantallas de distintos tamaños (escritorio, tabletas, móviles).


**Accesibilidad:** Verificación de contraste de colores para usuarios con daltonismo y tamaños de fuente legibles.

Este prototipo sirvió como base para validar el flujo de trabajo con stakeholders y realizar ajustes finales antes del desarrollo.





[Actividades](#) [Servicios](#) [Sobre nosotros](#) [Idioma](#) [Perfil](#)




**Jorge Serrano**

Correo@gmail.com

Inscribir un niño

Hijos

Lunes	Martes	Miercoles	Jueves	Viernes
Actividad 1	Actividad 1	Actividad 1	Actividad 1	Actividad 1
Actividad 2	Actividad 2	Actividad 2	Actividad 2	Actividad 2
Actividad 3	Actividad 3	Actividad 3	Actividad 3	Actividad 3
Actividad 4	Actividad 4	Actividad 4	Actividad 4	Actividad 4
Descanso	Descanso	Descanso	Descanso	Descanso
Actividad 6	Actividad 6	Actividad 6	Actividad 6	Actividad 6
Actividad 7	Actividad 7	Actividad 7	Actividad 7	Actividad 7
Actividad 8	Actividad 8	Actividad 8	Actividad 8	Actividad 8
Actividad 9	Actividad 9	Actividad 9	Actividad 9	Actividad 9



**Samuel Garcia**

Edad: 6 años  
Alergias: ninguna  
Grupo: A  
Tutor: Alex Sastre

Modificar



Sobre Nosotros

[Nuestro Blog](#)  
[Inscribirse](#)  
[Iniciar Sesión](#)

Legal





[Políticas](#)  
[Políticas de Cookies](#)  
[Política de privacidad](#)

Pagos seguros

Contacto

[Calle glori@Madridts](#)  
[458535744](#)  
[info@gmail.com](#)



## Planificación del Desarrollo

El proyecto se ejecutó siguiendo un enfoque ágil basado en sprints, diseñado para maximizar la flexibilidad y la adaptabilidad a los requisitos cambiantes del cliente. Cada sprint, con una duración promedio de dos semanas, culminaba en una entrega funcional que permitía validar avances y ajustar prioridades. A continuación, se desglosa el proceso en detalle:

### 1. Fase de Investigación y Definición de Requisitos

#### 1.1. Sesiones de descubrimiento con el cliente

Se organizaron talleres colaborativos con representantes del campamento (monitores) para:

Priorizar funcionalidades clave: calendario interactivo, sistema de notificaciones y paneles de control diferenciados por rol.

Se utilizaron técnicas como User Story Mapping para visualizar el flujo completo del usuario, desde el registro hasta la participación en actividades.

#### 1.2. Benchmarking competitivo

Se analizaron diferentes plataformas de otro campamento para ver sus fortalezas y debilidades:

Diseño UX/UI (ej.: eficiencia en formularios de inscripción).

Integración de tecnologías emergentes (como chatbots para consultas frecuentes).

### 2. Diseño de Arquitectura y Experiencia de Usuario

#### 2.1. Wireframing y estructuración

Se empleó Figma para crear wireframes interactivos, centrados en:

Homepage estratégica: Diseñada para convertir visitantes en usuarios registrados mediante llamados a acción destacados.

#### Flujos críticos:

Proceso de inscripción en 3 pasos con barra de progreso visible.

Acceso rápido a documentos requeridos (autorizaciones médicas, permisos legales).

#### 2.2. Guía de estilos detallada

Paleta cromática:

Color primario: Morado (#643A9B) para transmitir misterio.

Secundarios: Verde (#1ABC9C) para confianza y una conexión con la naturaleza.

Sistema de diseño:

Tipografía: Raleway para legibilidad en pantallas.

Iconografía: Set personalizado de 64 iconos en SVG con coherencia temática.

### 3. Desarrollo Técnico por Módulos

#### 3.1. Sprint 1: Módulos de autenticación y perfil de tutor

Tecnologías clave: JavaScript y PHP para frontend, MySQL para backend.

Funcionalidades implementadas:

Registro de tutores con validación en tiempo real (ej.: verificación de formato de correo electrónico u otras redes sociales).

Perfil personalizable: Para poder modificar los datos necesarios de los hijos para tener un mejor monitoreo.

#### 3.2. Sprint 2: Paneles de Control

##### Panel del Tutor:

Se implementó un sistema para visualizar información esencial de los niños, incluyendo datos médicos relevantes y calendarios personalizados.

Se integró un canal de comunicación seguro entre tutores y monitores, garantizando privacidad en los intercambios de información.

##### Panel del Monitor:

Se desarrolló un módulo para gestionar la asistencia diaria, combinando métodos tradicionales con tecnología innovadora para agilizar el proceso.

Se incluyeron herramientas automatizadas para generar reportes operativos, facilitando la documentación de actividades y eventos.

##### Panel del Administrador:

Se creó un centro de control centralizado con métricas en tiempo real, permitiendo tomar decisiones basadas en datos.

Se diseñó un sistema inteligente para organizar grupos de niños, optimizando criterios como edad, habilidades e intereses.

### 4. Base de Datos y Gestión de la Información

#### 4.1. Estructura y Optimización

Se definió un modelo de datos relacional para almacenar información de usuarios, actividades y registros históricos, asegurando relaciones lógicas entre entidades.

Se implementaron técnicas avanzadas para mejorar el rendimiento en consultas frecuentes, como la indexación estratégica de datos prioritarios.

#### 4.2. Seguridad y Cumplimiento

Todos los datos sensibles se protegieron mediante protocolos de

encriptación estándar, cumpliendo con normativas de protección de datos vigentes.

Se estableció un sistema de respaldo automático para prevenir pérdida de información, junto con protocolos de recuperación ante incidentes.

## **5. Despliegue y Operación**

### **5.1. Infraestructura y Escalabilidad**

La plataforma se alojó en un entorno escalable, capaz de adaptarse a fluctuaciones en la demanda de usuarios sin comprometer el rendimiento.

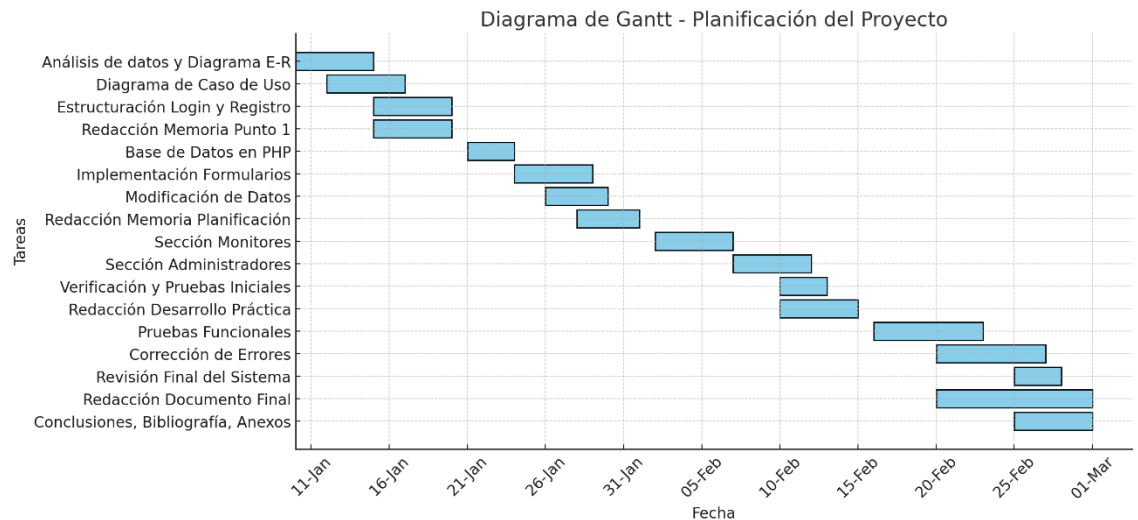
Se implementaron herramientas de monitoreo continuo para detectar y resolver incidencias proactivamente, asegurando alta disponibilidad.

### **5.2. Lanzamiento y Soporte Continuo**

El lanzamiento se realizó en fases, comenzando con un grupo piloto para validar estabilidad y usabilidad, seguido de una expansión gradual.

Se estableció un equipo técnico dedicado para brindar soporte post-implementación, junto con un ciclo de actualizaciones periódicas basadas en feedback de usuarios.

## Introducción al desarrollo



Se trata de una página web diseñada para que los padres puedan inscribir a sus hijos en un campamento de verano y gestionar sus necesidades durante su estancia.

También está diseñada para ayudar a los monitores del campamento a administrar las actividades de manera eficiente y comunicarse con los padres para cualquier comunicado.

Por último, contamos con una sección para el administrador, donde podrá gestionar las actividades, supervisar los pagos y resolver cualquier incidencia.

## 2. Funcionalidades implementadas

Explica cada una de las funcionalidades principales con detalle:

### Login funcional:

El sistema de login contará con dos botones que permitirán al usuario seleccionar su tipo de cuenta, ya sea padre, monitor o administrador. Una vez seleccionado el perfil, se mostrarán dos campos de texto donde deberá ingresar sus credenciales, incluyendo el correo electrónico y la contraseña.

El sistema validará los datos ingresados y, si son correctos, el usuario será redirigido automáticamente a su página de inicio personalizada, donde podrá acceder a las funciones correspondientes a su rol. En caso de error en las credenciales, se mostrará un mensaje indicando que los datos son incorrectos y ofreciendo la opción de recuperar la contraseña.

### Tutores:

Los tutores tienen acceso a diversas funcionalidades dentro de la plataforma que les permiten gestionar la inscripción de sus hijos, modificar datos personales y comunicarse con la administración del campamento.

Al acceder a la plataforma después de iniciar sesión, los tutores serán dirigidos a su página de inicio. Desde allí, podrán gestionar su perfil y realizar modificaciones en sus datos personales, como nombre, DNI, teléfono y otros detalles relevantes.

Además, en la misma página tendrán la opción de inscribir a sus hijos en el campamento completando un formulario. En este formulario podrán seleccionar el plan que mejor se adapte a sus necesidades y proporcionar información esencial del niño, como nombre, edad y datos opcionales como alergias u observaciones especiales.

Una vez inscritos, los tutores podrán visualizar la lista de sus hijos registrados mediante un desplegable y seleccionar a cada uno para gestionar su inscripción, permitiendo modificaciones en la información del niño si fuera necesario.

Otra funcionalidad importante es la comunicación dentro de la plataforma. Los tutores podrán enviar mensajes a la administración o a los monitores para resolver dudas, solicitar información o comunicar incidencias. Asimismo, tendrán la posibilidad de ver los mensajes recibidos para mantenerse informados sobre cualquier actualización, cambios en la programación o avisos importantes.

Con estas funcionalidades, la plataforma proporciona a los tutores un sistema eficiente y accesible para gestionar la participación de sus hijos en el campamento, asegurando un proceso de inscripción claro, una comunicación efectiva y la posibilidad de actualizar la información cuando sea necesario.

El menú de navegación también ofrecerá diversas funcionalidades, entre ellas:

- Calendario del campamento, donde podrán consultar las actividades programadas.
- Gestión de servicios, permitiendo solicitar opciones como el comedor o solicitar asesoramiento a los monitores mediante reuniones.
- Información del centro, con detalles sobre el equipo de monitores y las políticas del campamento.
- Cambio de idioma, dado que el centro es bilingüe, los usuarios podrán modificar el idioma de la plataforma en cualquier momento.
- Sección de notificaciones, donde podrán visualizar los mensajes enviados por los monitores y administradores con información relevante sobre sus hijos o el campamento.

### **Monitores:**

Los monitores tienen la responsabilidad de gestionar la asistencia de los niños. Esto incluye pasar lista en cada actividad o sesión del campamento, registrar ausencias y comunicar incidencias relacionadas con la asistencia. Este control permite un mejor seguimiento de la participación de cada niño y facilita la detección de posibles problemas.

Además, los monitores pueden supervisar los grupos asignados. Esto implica consultar la composición de cada grupo y gestionar la información relevante de cada niño dentro de su grupo. De esta manera, pueden asegurarse de que cada participante reciba la atención adecuada y se mantenga un ambiente organizado.

Otra función clave es la comunicación y el envío de notificaciones.

Los monitores pueden enviar avisos a los tutores sobre actividades, cambios en la programación o situaciones especiales. Asimismo, mantienen una comunicación fluida con la administración y otros monitores para coordinar tareas y resolver incidencias de manera eficiente.

**Admin:**

Los administradores contarán con diversas funcionalidades para gestionar de manera eficiente el campamento. Estas incluyen la planificación del calendario, la gestión de actividades, el control de pagos, la asignación de niños a grupos y la administración del equipo de monitores.

**Gestión de Niños**

Crear y gestionar el historial de cada niño dentro del campamento.

Asignar un niño a un grupo determinado y modificar su grupo en caso necesario.

Modificar el estado de pago de cada niño para reflejar si ha abonado la tarifa correspondiente.

**Administración de Actividades**

Crear nuevas actividades dentro del horario establecido.

Validar mediante JavaScript que las actividades creadas estén dentro del intervalo de tiempo del plan al que pertenecen.

**Gestión de Planes de Fechas**

Crear planes de fechas detallados con los días y horarios del campamento.

Eliminar planes en caso de que ya no sean necesarios.

**Administración de Pagos**

Revisar el estado de los cobros de cada participante.

Gestionar posibles incidencias o retrasos en los pagos.

Asegurar que todos los participantes estén correctamente registrados y al día con sus tarifas.

**Gestión del Equipo de Monitores**

Contratar y gestionar monitores dentro de la plataforma.

Asignarles roles y definir sus responsabilidades dentro del campamento.

Garantizar que cada monitor tenga acceso a las herramientas necesarias para desempeñar sus funciones.

**3. Arquitectura del sistema****Explicación de cómo están organizados los archivos y módulos del proyecto.**

En cuanto a la organización del archivo en la que se desarrolla el proyecto, se ha dividido en diferentes carpetas en su interior para facilitar la gestión y estructuración del código.

La carpeta principal del proyecto contiene una subcarpeta denominada CODE, en la cual se encuentran todos los archivos

esenciales para el desarrollo. Dentro de esta carpeta se organizan los archivos de los distintos lenguajes utilizados, incluyendo HTML, CSS, JavaScript, PHP, SQL y una carpeta adicional denominada assets, donde se almacenarán recursos como imágenes, fuentes y otros elementos necesarios para el diseño y funcionamiento del proyecto.

Dentro de la carpeta CODE, se implementa la estructura basada en el modelo Vista-Controlador (MVC). Este modelo permite una mejor distribución y organización de los archivos, separando claramente la lógica del negocio, la presentación y el control del flujo de datos. De este modo, se logra una mayor escalabilidad y mantenimiento del proyecto, evitando la mezcla de código y facilitando futuras modificaciones o expansiones.

Para la comunicación entre el frontend y el backend, se emplean tecnologías como Fetch API y el formato de datos JSON. La implementación de estas tecnologías permite el intercambio de información de manera eficiente, asegurando una correcta transmisión de datos entre la interfaz de usuario y el servidor. Gracias a esta integración, es posible enviar y recibir datos en tiempo real, mejorar la experiencia del usuario y optimizar el rendimiento del sistema.

Con esta organización y metodología, el proyecto se desarrolla de manera estructurada, asegurando claridad en el código y facilitando el trabajo colaborativo dentro del equipo de desarrollo.

#### **4. Ejecución del sistema**

La web estará diseñada para ser completamente funcional en cualquier dispositivo, ya sea en ordenadores, teléfonos móviles. Su diseño responsivo garantizará una experiencia de usuario óptima, adaptándose automáticamente a diferentes tamaños de pantalla y resoluciones. Esto permitirá a los usuarios acceder a la plataforma de manera cómoda y fluida desde cualquier lugar y en cualquier momento, sin importar el dispositivo que utilicen.

Para mejorar la eficiencia y la rapidez en el acceso, la base de datos se cargará automáticamente al iniciar la pantalla de login. De esta manera, cuando el usuario introduzca sus credenciales, el sistema podrá validar los datos de forma inmediata y permitir el acceso sin interrupciones ni retrasos, siempre que la información proporcionada sea correcta. Este proceso garantizará una experiencia más ágil y fluida, evitando tiempos de espera innecesarios o posibles errores de conexión con la base de datos.

Además, la plataforma contará con optimizaciones en la carga de datos para asegurar un rendimiento eficiente, incluso en conexiones de internet más lentas, mejorando así la accesibilidad y la usabilidad para todos los usuarios.

#### **5. Pruebas y validaciones**



El sistema ha sido sometido a distintas pruebas para garantizar su correcto funcionamiento y detectar posibles errores antes de su implementación final. Se han realizado los siguientes tipos de pruebas:

**Pruebas unitarias:** Se han llevado a cabo pruebas automatizadas en funciones clave del sistema, como la validación de formularios, la autenticación de usuarios y la conexión con la base de datos. Estas pruebas han permitido verificar que cada componente funcione correctamente de forma independiente.

**Pruebas manuales:** Se han realizado pruebas funcionales navegando por la plataforma como un usuario real, verificando la correcta interacción con los formularios, botones, menús y demás elementos de la interfaz.

**Pruebas de integración:** Se ha comprobado que la comunicación entre el frontend y el backend sea fluida, validando que los datos enviados desde los formularios se almacenen correctamente en la base de datos y que las consultas devuelvan la información adecuada.

#### **Casos de prueba importantes**

Se han identificado y probado diversos escenarios clave para asegurar la estabilidad del sistema:

##### **Inicio de sesión con credenciales incorrectas:**

Se ha probado qué sucede cuando un usuario introduce un correo o contraseña incorrectos, verificando que el sistema muestre un mensaje de error adecuado y no permita el acceso.

##### **Registro con datos inválidos:**

Se han probado intentos de registro con datos incompletos o erróneos (como correos con formato incorrecto, contraseñas demasiado cortas o campos vacíos) para comprobar que el sistema muestre alertas y evite el registro.

##### **Inscripción de un niño al campamento:**

Se ha probado el proceso de inscripción para asegurarse de que los datos introducidos se almacenen correctamente y que el usuario pueda ver y modificar la información después de guardarla.

##### **Carga de la base de datos al iniciar sesión:**

Se ha comprobado que, al introducir credenciales correctas, la información del usuario se cargue sin demoras ni errores, garantizando un acceso rápido y sin interrupciones.

##### **Prueba de cambios en el perfil del usuario:**

Se ha probado la modificación de datos personales, asegurando que los cambios se reflejen correctamente en la base de datos y en la interfaz.

##### **Compatibilidad con diferentes dispositivos y navegadores:**

Se han realizado pruebas en distintos navegadores (Chrome, Firefox, etc...) y en dispositivos de diferentes tamaños para garantizar un diseño responsivo y una navegación fluida.

## Conclusiones

En conclusión, este proyecto ha logrado integrar diversas áreas de la programación, abarcando tanto el desarrollo frontend como backend, con el propósito de crear una página web funcional y eficiente para los usuarios de un campamento de verano. Se han implementado elementos tanto dinámicos como estáticos, lo que permite una experiencia de navegación fluida e interactiva. Además, se ha prestado especial atención al diseño y la estética de la página, asegurando una interfaz atractiva, intuitiva y accesible para los usuarios. La combinación de estas tecnologías y enfoques ha dado como resultado un sitio web bien estructurado, que no solo cumple con los requisitos funcionales del campamento, sino que también ofrece una experiencia de usuario óptima y visualmente agradable.

A pesar del éxito del desarrollo, siempre existen oportunidades de mejora y expansión. Uno de los aspectos a considerar en futuras versiones del proyecto es la optimización del rendimiento y escalabilidad. Se podrían implementar técnicas como la carga diferida para mejorar la velocidad de carga de los recursos, especialmente imágenes y archivos multimedia. Además, la optimización de consultas a la base de datos mediante índices y procedimientos almacenados podría reducir significativamente los tiempos de respuesta. En términos de arquitectura, explorar un modelo basado en microservicios facilitaría la escalabilidad del proyecto y su mantenimiento a largo plazo.

Otro punto clave a mejorar es la seguridad y protección de datos. Para fortalecer la autenticación de los usuarios, sería recomendable implementar un sistema de autenticación multifactor (MFA), especialmente para perfiles sensibles como administradores y monitores. Asimismo, el uso de cifrado avanzado en el almacenamiento de datos personales minimizaría riesgos de vulnerabilidades ante posibles ataques. También es fundamental mantener actualizadas todas las librerías y dependencias del proyecto para evitar fallos de seguridad derivados de versiones obsoletas.

En cuanto a la ampliación de funcionalidades, se podría integrar un sistema de chat en tiempo real que facilite la comunicación entre tutores y monitores, optimizando la interacción dentro de la plataforma. Además, la implementación de notificaciones push permitiría mantener a los usuarios informados sobre eventos relevantes del campamento. Otra mejora sería la integración de herramientas de analítica web, que ayudarían a evaluar el comportamiento de los usuarios y a realizar ajustes en la interfaz para mejorar la usabilidad de la plataforma.

Por último, la experiencia del usuario (UX/UI) también podría mejorarse con diversas estrategias. Realizar pruebas de usabilidad con usuarios reales permitiría identificar posibles

puntos de fricción en la navegación. Asimismo, la incorporación de un modo oscuro brindaría una mejor accesibilidad y comodidad visual en distintas condiciones de iluminación. Además, priorizar un enfoque mobile-first garantizaría una mejor optimización en dispositivos móviles, asegurando que la plataforma sea completamente funcional en cualquier pantalla.

En resumen, el proyecto ha cumplido con los objetivos iniciales, proporcionando una solución funcional y eficiente para la gestión del campamento. Sin embargo, el desarrollo web es un proceso continuo, y la implementación de mejoras futuras garantizará una experiencia aún más completa y satisfactoria para los usuarios.

## Bibliografía

### Manuales utilizados:

Mozilla Developer Network. Documentación de JavaScript, HTML y CSS. Disponible en: <https://developer.mozilla.org>

PHP Documentation. Manual de PHP. Disponible en: <https://www.php.net/docs.php>

MySQL Documentation. Guía de MySQL. Disponible en: <https://dev.mysql.com/doc/>

### Recursos digitales utilizados:

#### Figma:

Autor: Evan Wallace y Dylan Field

Creación: 22 sept 2022

<https://www.figma.com/login>

#### Visual Studio Code:

Autor: Microsoft

Creación: 29 de abril de 2015

<https://code.visualstudio.com/>

#### Xamp:

Autor: Apache Friends

Creación: 4 de septiembre de 2002

<https://www.apachefriends.org/es/index.html>

#### Canva:

Autor: Melanie Perkins, Cliff Obrecht y Cameron Adams

Creación: 1 de enero de 2012

<https://www.canva.com/>

#### PHPMyadmin:

Autor: Olivier Müller, Marc Delisle y Loïc Chapeaux

Creación: diciembre del 2002

<https://www.phpmyadmin.net/>

#### Github:

Autor: Tom-Preston-Werner Chris Wanstrath P. J.

Hyett Scott Chacon

Creación: 8 de febrero de 2008

<https://github.com/>

### Referencias de otros campamentos:

gmrcamps: <https://www.gmrcamps.es/>

### Videos utilizados:

#### Youtube:

Autor: Vector M:

**Publicación: 29 ago 2023**

**Login:**

<https://youtu.be/Uhbn1KmiNbg?si=4YWR4h9GGMco5bBe>

**Tarjetas:**

**Autor: Coder Techy**

**Publicación: 20 may 2023**

[https://youtube.com/shorts/zs4L2xMWcz0?si=qLin\\_K69Am7l4DAAd](https://youtube.com/shorts/zs4L2xMWcz0?si=qLin_K69Am7l4DAAd)

**Autor: Bedimcode**

**Publicación: 2 oct 2023**

[https://youtu.be/Q8BamnhOmWc?si=LySQOVf4MtnAx\\_vO](https://youtu.be/Q8BamnhOmWc?si=LySQOVf4MtnAx_vO)

## Anexos:

### Pruebas manuales del PHP

Aquí tenemos la estructuración del formulario para rellenar en el login

```

<div class="main">
    <input type="checkbox" id="chk" aria-hidden="true">
    <!-- Formulario de Registro -->
    <div class="signup">
        <form id="signupForm">
            <label for="chk" aria-hidden="true">Crear Cuenta</label>
            <input type="email" name="email" placeholder="Email" id="reg_email" required>
            <input type="password" name="pswd" placeholder="Password" id="reg_pswd" required>
            <button type="submit">Crear Cuenta</button>
        </form>
    </div>

    <!-- Formulario de Login -->
    <div class="login">
        <form id="loginForm">
            <label for="chk" aria-hidden="true">Login</label>
            <input type="email" name="email" placeholder="Email" id="email" required>
            <input type="password" name="pswd" placeholder="Password" id="pswd" required>
            <button type="submit">Login</button>
        </form>
    </div>
</div>
<script src="../JS/operarLogin.js"></script>
<script src="../JS/operarRegistro.js"></script>
</body>
</html>
    
```

En esta parte se ve en donde se realizan las comprobaciones de los datos que se introducen en la parte del login, dependiendo del tipo de usuario





```

//-----//
// Escapar los datos para prevenir SQL Injection
//-----//
$email = $conn->real_escape_string(string: $email);
$password = $conn->real_escape_string(string: $password);

//-----//
// Verificar en la tabla TUTORES (Padres)
//-----//
$sql = "SELECT id_tutor, contrasenia FROM TUTORES WHERE email = '$email'";
$result = $conn->query(query: $sql);
if ($result === false) {
    manejarError(mensaje: "Error en la consulta SQL: " . $conn->error);
}
if ($result && $result->num_rows > 0) {
    $row = $result->fetch_assoc();
    // Verificar si la contraseña ingresada coincide con la hasheada en la BD
    if (password_verify(password: $password, hash: $row['contrasenia'])) {
        $_SESSION["usuario"] = $email;
        $_SESSION["tipo"] = "TUTOR";
        $_SESSION["id"] = $row["id_tutor"];
        $_SESSION["login"] = "OK";
        echo json_encode(value: ["redirect" => "../html/IndexPadre.html"]);
        exit();
    }
}
}
    
```

Aquí se puede ver reflejado la base de datos con los usuarios existentes

Opciones extra

		id_tutor	nombre	dni	telefono	email	contrasenia
<input type="checkbox"/>	 Editar  Copiar  Borrar	1	Tutor Ejemplo	12345678A	123456789	tutor@ejemplo.com	\$2y\$10\$Bd77ajvlvw8G4ExY0CQ57ewHfT2ctoziqRgpYvoF4Q
<input type="checkbox"/>	 Editar  Copiar  Borrar	2				hanlizhao888@gmail.com	\$2y\$10\$OGUW1CLT4rGfO7HRJcTCG.oCkYpwhdVzkSe5MAj

Se establece la conexión con la base de datos

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
  <link rel="stylesheet" href="../css/LoginStyle.css">
  <title>Iniciar Sesión</title>
  <link href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap" rel="stylesheet">
</head>
<body>
  <div class="logo">
    
  </div>
  <script>
    // Ejecutar Conexion.php cuando la página se carga
    window.onload = function() {
      fetch("../Server/Conexion.php")
        .then(response => response.text())
        .then(data => console.log(data)) // Muestra los mensajes en la consola
        .catch(error => console.error("Error al ejecutar Conexion.php:", error));
    };
  </script>
```

Mensajes de error si los campos no son correctos

The image shows a login form titled "Login" with a "Crear Cuenta" link. The form has two input fields: "Email" and "Password". Below the "Email" field, there is a red error message: "El correo no puede estar vacío." Below the "Password" field, there is a red error message: "La contraseña no puede estar vacía." At the bottom of the form is an orange "Login" button. The form is set against a background image of a sunset over a body of water with a yellow tent in the foreground.

Si los campos son los correctos nos permitirá acceder a nuestro index dependiendo del usuario

The image shows a web application interface for 'Wooky Dooky Camp'. The top section is a login and registration form with a blue background. It features the 'Wooky Dooky Camp' logo in the top left corner, which depicts a cartoon dog wearing a cowboy hat. The form has two main sections: 'Crear Cuenta' (Create Account) and 'Login'. The 'Login' section contains two input fields: one for the email address 'hanlizhao888@gmail.com' and another for a password represented by dots. Below these fields is an orange 'Login' button. The bottom section of the image shows a navigation bar with links: 'Actividades', 'Servicios', 'Sobre nosotros', 'Idiomas', and 'Perfil'. Below the navigation bar is a large gray box containing the 'Wooky Dooky Camp' logo and a welcome message for tutors: 'Bienvenido a nuestra interfaz para los tutores'. Below this message is a paragraph of placeholder text: 'Este es el texto de la introducción. Aquí puedes agregar más detalles sobre el contenido que deseas mostrar.'



También están las validaciones de los formularios de inscripción, con su mensajes de error correspondientes.

## Inscripción del campamento

### Datos del representante legal

Nombre completo

 El nombre no puede estar vacío


Correo electrónico

 El correo no puede estar vacío

Teléfono

 El teléfono no puede estar vacío

DNI / NIF

 El DNI no puede estar vacío

```

Tabnine | Edit | Test | Explain | Document
function comprobarNombre () {
  if (nombre.value == "") {
    mostrarError(errornombre, "El nombre no puede estar vacío");
  } else {
    mostrarError(errornombre, "");
  }
}
}
Tabnine | Edit | Test | Explain | Document
function comprobarCorreo() {
  if (correo.value == "") {
    mostrarError(errorcorreo, "El correo no puede estar vacío");
  } else {
    mostrarError(errorcorreo, "");
    if (/^[A-Za-z0-9\u4e00-\u9fa5]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+$/ .test(correo.value)){
      mostrarError(errorcorreo, "");
    } else {
      mostrarError(errorcorreo, "Error de formato de Correo, tiene que contener un @");
    }
  }
}
}
Tabnine | Edit | Test | Explain | Document
function comprobarTelefono() {
  if (telefono.value == "") {
    mostrarError(errortelefono, "El teléfono no puede estar vacío");
  } else {
    mostrarError(errortelefono, "");
    if (/^[0-9]{9}$/ .test(telefono.value)){
      mostrarError(errortelefono, "");
    } else {
      mostrarError(errortelefono, "El teléfono tiene que ser de 9 dígitos de número");
    }
  }
}
}

```

Aquí podemos ver cómo se crea la base de datos y se crean las tablas que lo conforman:

```

1  <?php  MRYIVAN, hace 2 semanas • Add files via upload
2  //-----//
3  // Configuración de conexión
4  //-----//
5  $servername = "localhost";
6  $username = "root";
7  $password = "";
8  $dbname = "Campamento";
9
10 //-----//
11 // Crear la conexión (sin seleccionar la base aún)
12 //-----//
13 $conn = new mysqli(hostname: $servername, username: $username, password: $password);
14 if ($conn->connect_error) {
15     die("Connection failed: " . $conn->connect_error);
16 }
17
18 //-----//
19 // Crear la base de datos si no existe
20 //-----//
21 $sql = "CREATE DATABASE IF NOT EXISTS $dbname";
22 $conn->query(query: $sql); // Se crea la base de datos si no existe, sin imprimir nada
23
24 //-----//
25 // Seleccionar la base de datos
26 //-----//
27 $conn->select_db(database: $dbname);
28
29 //-----//
30 // Creación de tablas
31 //-----//
32 $sql_tables = "
33
34 CREATE TABLE IF NOT EXISTS PLAN_FECHAS (
35     id_plan INT PRIMARY KEY AUTO_INCREMENT,
36     nombre VARCHAR(40) NOT NULL,
37     fecha_inicio DATE NOT NULL,
38     fecha_fin DATE NOT NULL,
39     fecha_maxInscripcion DATE NOT null,
40     hora_maximaInscripcion time not null,
41     precio VARCHAR(9) NOT NULL,
42     definicion VARCHAR(40000) NOT NULL
43 );
44
45 CREATE TABLE IF NOT EXISTS TUTORES (
46     id_tutor INT PRIMARY KEY AUTO_INCREMENT,
47     nombre VARCHAR(50) NOT NULL,
48     dni VARCHAR(9) NOT NULL,
49     telefono VARCHAR(9) NOT NULL,
50     email VARCHAR(50) NOT NULL UNIQUE,
51     contrasenia text NOT NULL,
52     avatar_src text
53 );
54
55 CREATE TABLE IF NOT EXISTS PLAN_COMEDOR (
56     id_plan_comedor INT PRIMARY KEY AUTO_INCREMENT,
57     nombre_plan VARCHAR(100) NOT NULL,
58     descripcion TEXT NOT NULL,
59     precio DECIMAL(10, 2) NOT NULL,
60     imagenComida_src text
61 );
62
63 CREATE TABLE IF NOT EXISTS NINOS (
64     id_nino INT PRIMARY KEY AUTO_INCREMENT,
65     nombre VARCHAR(50) NOT NULL,

```

Aquí podemos ver cómo se crean las cuantas, de los diferentes usuarios, con las contraseñas cifradas con insert en las tablas con sus validaciones

```

168 // ----- 1. MONITORES -----
169 // -----
170 // Funcion para crear datos en la tabla monitores con un id asignado ademas de su contraseña "hashe
171 // -----
172 if (!usuarioExiste(conn: $conn, tabla: "MONITORES", email: "monitor@ejemplo.com")) {
173     $stmt = $conn->prepare(query: "INSERT INTO MONITORES (nombre, email, contrasenia, descripcion) \
174     $nombre = "Monitor Ejemplo";
175     $email = "monitor@ejemplo.com";
176     $hashed_password = '$2y$10$BdT7ajv1vw8G4ExY0CQ57ewHFT2ctoziqRgpYvoF4QA41uu0/VEgu';
177     $descripcion = "Monitor de ejemplo para el campamento.";
178     $stmt->bind_param(types: "ssss", var: &$nombre, vars: &$email, $hashed_password, $descripcion);
179
180     if ($stmt->execute()) {
181         echo "Monitor insertado correctamente.<br>";
182     } else {
183         echo "Error al insertar Monitor: " . $stmt->error . "<br>";
184     }
185     $stmt->close();
186 }
187
188 // ----- 3. TUTORES -----
189 // -----
190 // Insertar Tutor si no existe en la base de datos.
191 // -----
192 if (!usuarioExiste(conn: $conn, tabla: "TUTORES", email: "tutor@ejemplo.com")) {
193     $stmt = $conn->prepare(query: "INSERT INTO TUTORES (nombre, dni, telefono, email, contrasenia) \
194     $nombre = "Tutor Ejemplo";
195     $dni = "12345678A";
196     $telefono = "123456789";
197     $email = "tutor@ejemplo.com";
198     $tutor_hashed_password = '$2y$10$BdT7ajv1vw8G4ExY0CQ57ewHFT2ctoziqRgpYvoF4QA41uu0/VEgu';
199     $stmt->bind_param(types: "sssss", var: &$nombre, vars: &$dni, $telefono, $email, $tutor_hashed_p
200

```

Aquí en la parte de JavaScript vemos la implementación de un chat para q los tutores puedan contactar con los monitores.

Este código en jQuery gestiona la interacción con un menú y la carga dinámica de una lista de monitores mediante AJAX. Primero, espera a que el documento esté listo y agrega un evento para mostrar u ocultar el menú cuando se hace clic en #action\_menu\_btn. Luego, realiza una petición POST a "GestionarNotificacionesPadre.php", solicitando la lista de monitores en formato JSON. Si la respuesta es exitosa, limpia y actualiza dinámicamente la lista de contactos con los datos recibidos, mostrando el nombre y el avatar del monitor (o una imagen predeterminada si no hay avatar disponible). En caso de error, muestra un mensaje en la consola. Finalmente, agrega un evento delegado para detectar clics en los elementos de la lista, permitiendo resaltar el monitor seleccionado y actualizar la información en la interfaz.

```
$(document).ready(function(){
    // Toggle del menú de acción
    $('#action_menu_btn').click(function(){
        $('.action_menu').toggle();
    });

    // Realizamos la petición AJAX para obtener los monitores
    $.ajax({
        url: "../Server/GestionarNotificacionesPadre.php", // Asegúrate de que la ruta es correcta
        type: "POST",
        data: { accion: "obtener_monitores" },
        dataType: "json",
        success: function (data) {
            // Selecciona el contenedor de la lista de contactos
            var contactsList = $(".contacts");
            contactsList.empty(); // Limpia cualquier contenido previo

            if (data.length > 0) {
                data.forEach(function (monitor) {
                    var avatar = monitor.avatar_src && monitor.avatar_src.trim() !== ""
                        ? monitor.avatar_src
                        : "../assets/img/avatar.png";
                    // Genera el HTML para cada monitor, agregando data-nombre
                    var monitorHTML = `
                    <li data-id="${monitor.id_monitor}" data-nombre="${monitor.nombre}">
                    <div class="d-flex bd-highlight">
                    <div class="img_cont">
                    
                    <span class="online_icon"></span>
                    </div>
                    <div class="user_info">

```

```

        <span class="online_icon"></span>
      </div>
      <div class="user_info">
        <span>${monitor.nombre}</span>
      </div>
    </div>
  </li>
  `;
  contactsList.append(monitorHTML);
});
} else {
  contactsList.append("<li>No hay monitores</li>");
}
},
error: function (xhr, status, error) {
  console.error("Error al obtener los monitores:", error);
}
});

// Delegar el evento de clic en los elementos <li> de la lista de contactos
$(document).on("click", ".contacts li", function(){
  var nombre = $(this).data("nombre");
  // Remueve la clase active de todos los elementos
  $(this).siblings().removeClass('active');
  // Agrega la clase active al elemento clickeado
  $(this).addClass('active');
  // Actualiza el indicador de usuario con el nombre del monitor seleccionado
  $(".card-header.msg_head .user_info").html("<span>" + nombre + "</span>");
});

```

## Instalación de apache

Para poder instalar, utilizaremos dos comandos

*\$sudo apt-get update*

*\$sudo apt-get install apache2*

```
Configurando apache2-data (2.4.58-1ubuntu8.4) ...
Configurando libaprutil1:amd64 (1.6.3-1.1ubuntu7) ...
Configurando libaprutil1-ldap:amd64 (1.6.3-1.1ubuntu7) ...
Configurando libaprutil1-dbd-sqlite3:amd64 (1.6.3-1.1ubuntu7) ...
Configurando apache2-utils (2.4.58-1ubuntu8.4) ...
Configurando apache2-bin (2.4.58-1ubuntu8.4) ...
Configurando apache2 (2.4.58-1ubuntu8.4) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Procesando disparadores para ufw (0.36.2-6) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
Procesando disparadores para libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@wjxant-server:/home/wjxant#
```

Podemos comprobar que se ha creado un nuevo directorio,  
podemos comprobar con el siguiente código:

*\$cd /var/www/html/*

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@wjxant-server:/home/wjxant# cd /var/www/html/
root@wjxant-server:/var/www/html# ls
index.html
root@wjxant-server:/var/www/html# ls -l
total 12
-rw-r--r-- 1 root root 10671 oct  9 07:37 index.html
root@wjxant-server:/var/www/html# _
```

Para escuchar los apaches activos utilizamos:

```
root@wjxant-server:/var/www/html# netstat -ltn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
root@wjxant-server:/var/www/html#
```

Ahora comprobamos el contenido del */var/www/html*

```
wjxant@wjxant-server:/var/www/html$ ls
index.html
```

Podemos ver que en el directorio html hay un archivo que se llama index.html. Con el nano podemos ver que es un archivo html

```

GNU nano 7.2 index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
  Modified from the Debian original for Ubuntu
  Last updated: 2022-03-22
  See: https://launchpad.net/bugs/1966004
-->
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Apache2 Ubuntu Default Page: It works</title>
  <style type="text/css" media="screen">
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}

body, html {
  padding: 3px 3px 3px 3px;

  background-color: #D8DBE2;

  font-family: Ubuntu, Verdana, sans-serif;
  font-size: 11pt;
  text-align: center;
}

div.main_page {
  position: relative;
  display: table;

  width: 800px;

  margin-bottom: 3px;
  margin-left: auto;
  margin-right: auto;
  padding: 0px 0px 0px 0px;

  border-width: 2px;
  border-color: #212738;
  border-style: solid;

  background-color: #FFFFFF;

  text-align: center;
}
  
```

[ File 'index.html' is unwritable ]

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-I To Bracket M-O Prev  
 Exit Read File Replace Paste Justify Go To Line M-F Redo M-E Copy M-W Where Was M-N Next

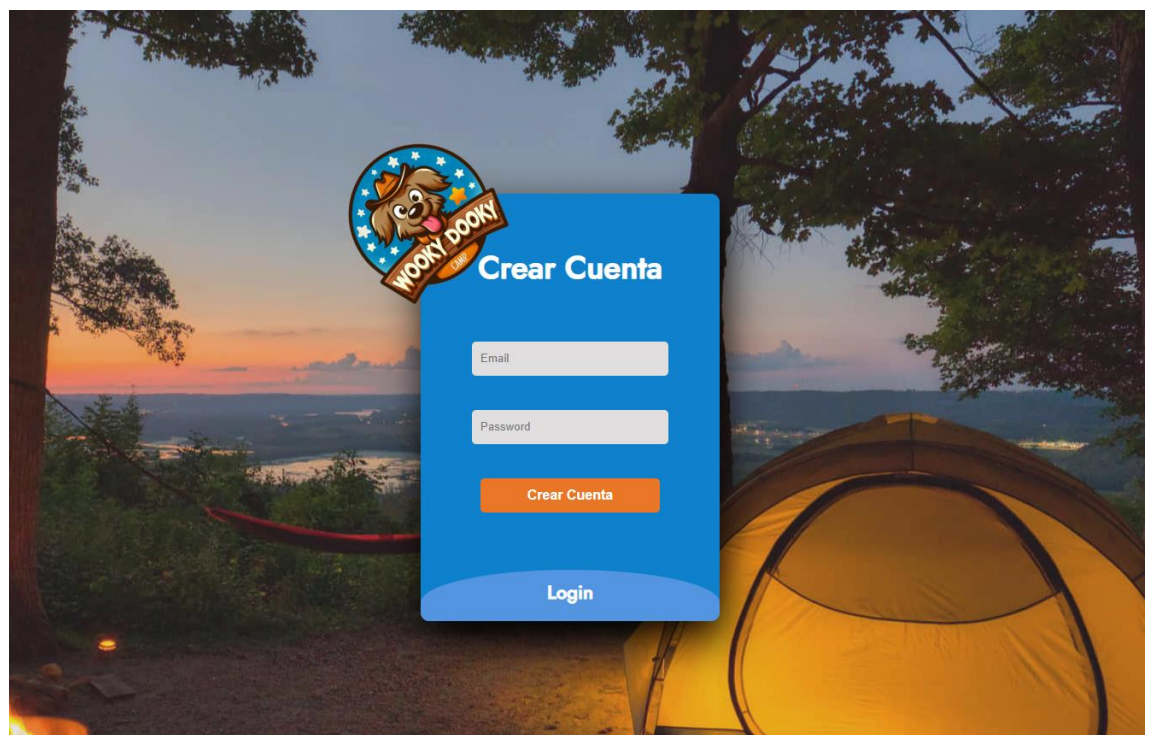


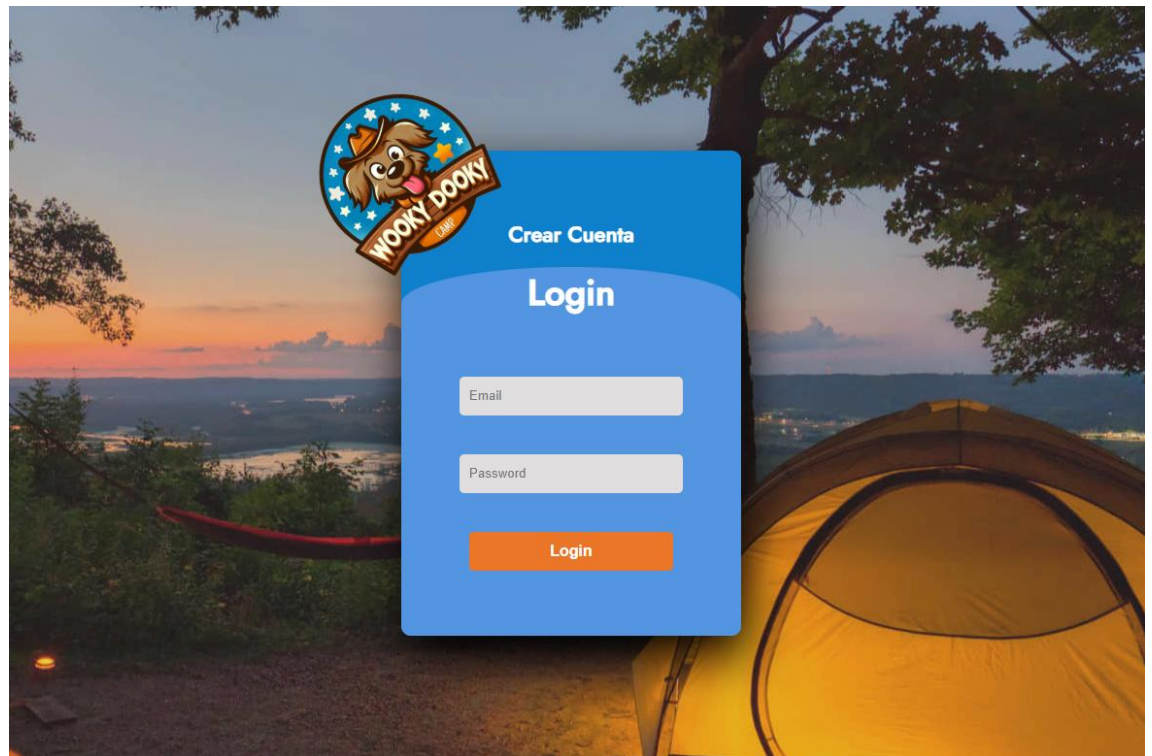
## Manuar de usuario

Inicialmente el usuario tendrá que entrar en nuestra pagina web para poder llegar a la opción de iniciar sesión:



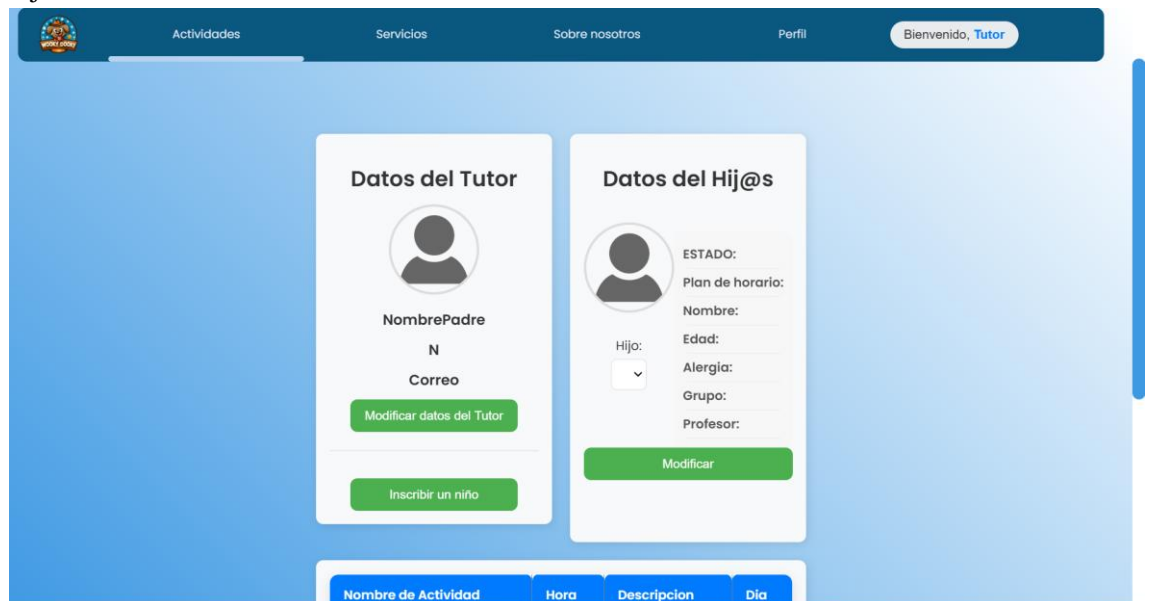
Tras ello tendrá que rellenar los campos correspondientes o crear una cuenta nueva si es un nuevo usuario:





## Manual para Padres - Guía de Uso y Funcionalidades

Una vez haya iniciado sesión, las opciones disponibles dependerán del tipo de usuario que corresponda a su perfil. Si es tutor legal, contará con diversas funcionalidades pensadas para mejorar la experiencia y el seguimiento de la actividad de sus hijos.



### Opciones para el Tutor Legal

Como tutor legal, tendrá acceso a una serie de funciones que le permitirán gestionar de manera eficiente la información de sus hijos dentro del centro.

### 1. Gestión del Perfil

Desde la sección de perfil, podrá actualizar su información personal, asegurándose de que sus datos de contacto, dirección y cualquier otro detalle relevante estén siempre actualizados. Esto es importante para recibir notificaciones y mantenerse informado sobre cualquier comunicación del centro.

### 2. Inscripción y Gestión de Hijos

Tendrá la posibilidad de inscribir a sus hijos en las actividades ofrecidas por el centro, asegurándose de que participen en aquellas que sean de su interés. Además, podrá modificar la información de sus hijos en caso de que sea necesario actualizar datos personales, información médica relevante o cualquier otro detalle importante.

Además de la gestión de perfil e inscripción, la plataforma cuenta con diversas herramientas para mejorar la experiencia y la comunicación dentro del centro.

## Inscripción del campamento

### Datos del representante legal

Nombre completo

Correo electrónico

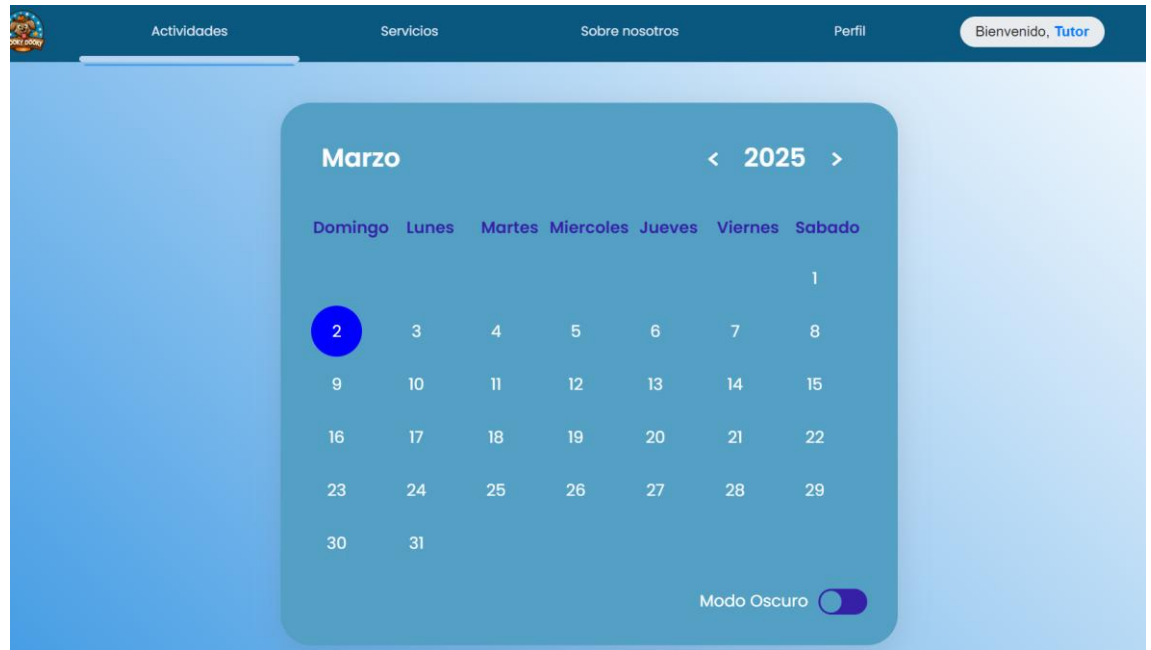
Teléfono

DNI / NIF

### 3. Calendario de Actividades

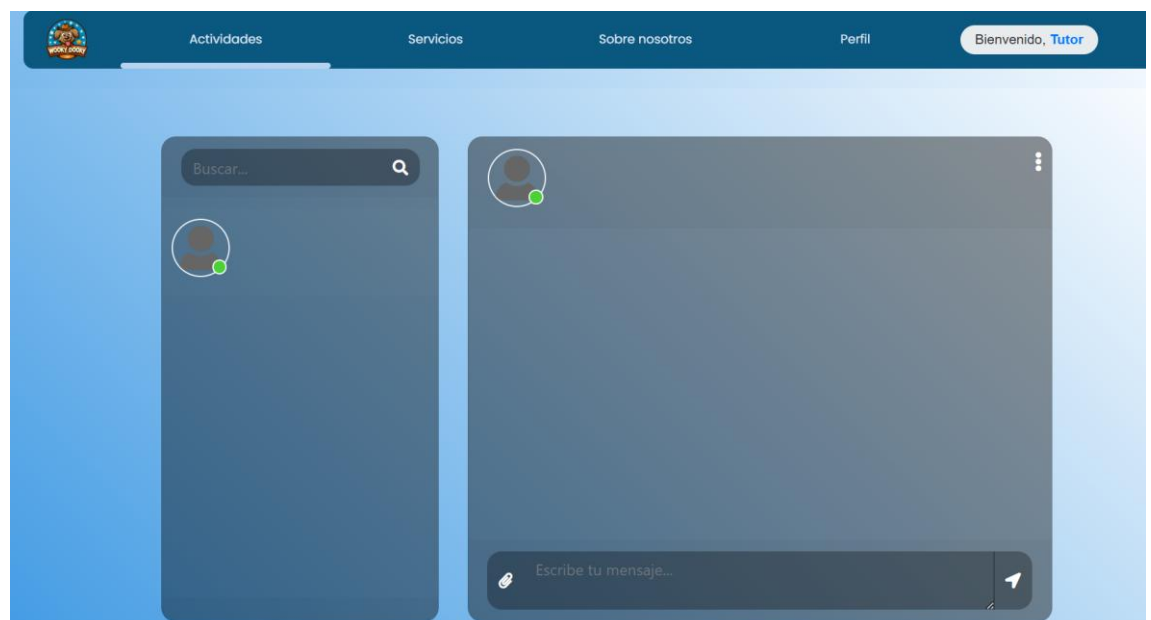
A través del calendario interactivo, podrá consultar todas las actividades programadas en el centro. Esto le permitirá estar al tanto de eventos, fechas importantes y actividades en las que participarán sus hijos, facilitando la organización del tiempo y

la planificación familiar.



#### 4. Chat con Monitores

Para mejorar la comunicación, la plataforma ofrece un sistema de mensajería que permite a los tutores legales contactar directamente con los monitores responsables de sus hijos. Esta herramienta facilita la resolución de dudas, consultas sobre el desempeño de los niños y cualquier otro tipo de información relevante.



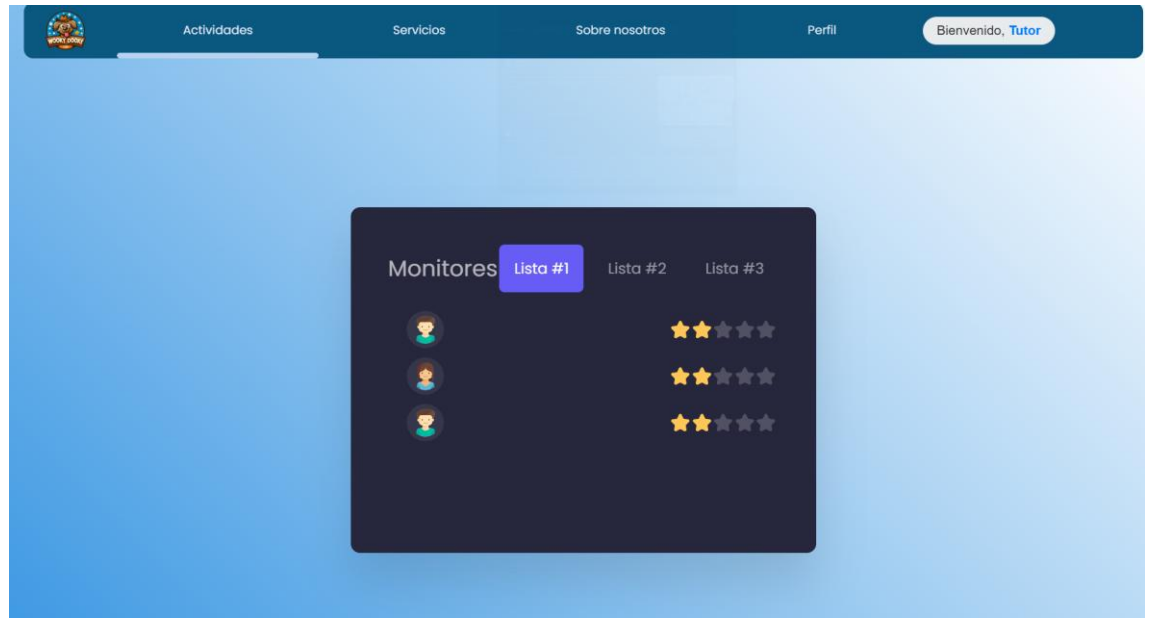
#### 5. Listado de Actividades

Podrá acceder a un listado detallado con todas las actividades disponibles en el centro. Cada actividad incluirá información sobre los horarios, edades recomendadas, requisitos y monitores

encargados, permitiéndole elegir aquellas que mejor se adapten a las necesidades e intereses de sus hijos.

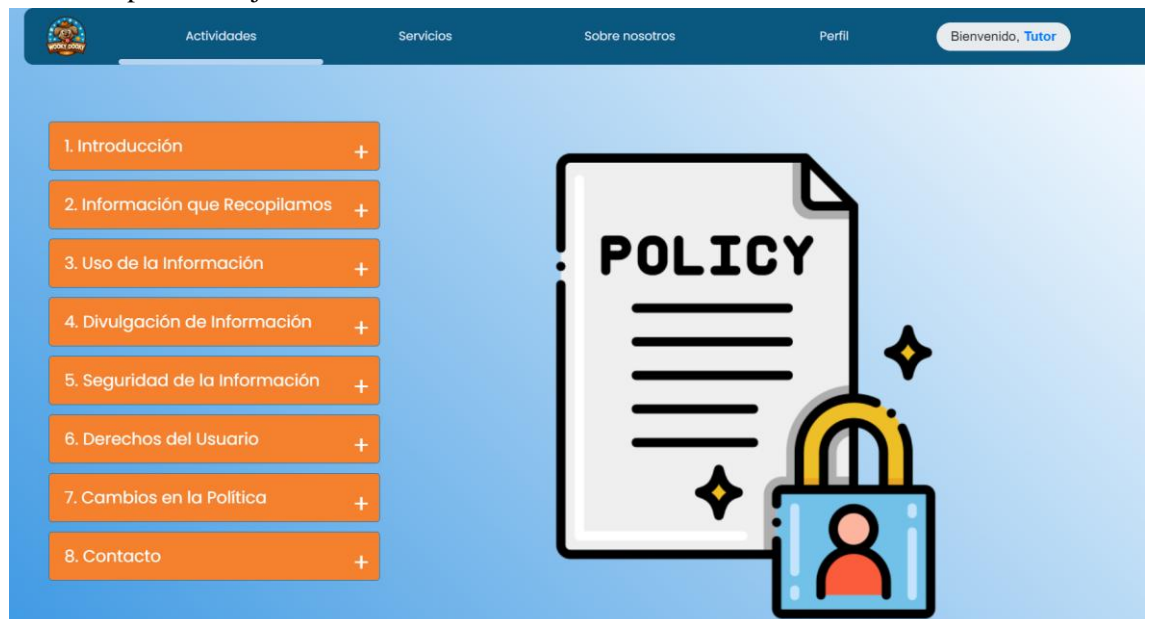
## 6. Información sobre los Monitores

La plataforma permite consultar información sobre los monitores del centro, incluyendo sus nombres. Esto le ayudará a conocer mejor a las personas que estarán a cargo de sus hijos durante su estancia en el centro.



## 7. Políticas del Centro

En esta sección, podrá consultar todas las normativas y políticas del centro, incluyendo reglamentos internos, normas de convivencia, protocolos de seguridad y cualquier otra información relevante para garantizar un entorno seguro y adecuado para sus hijos.



## 8. Cierre de Sesión



Para garantizar la seguridad de su cuenta y la privacidad de la información, se recomienda cerrar sesión una vez haya terminado de utilizar la plataforma, especialmente si accede desde un dispositivo compartido.

## Manual para Monitores - Guía de Uso y Funcionalidades

Una vez hayas iniciado sesión, podrás acceder a las siguientes funcionalidades, pensadas para mejorar la eficiencia en la gestión de tu trabajo diario.

### Opciones para el Monitor

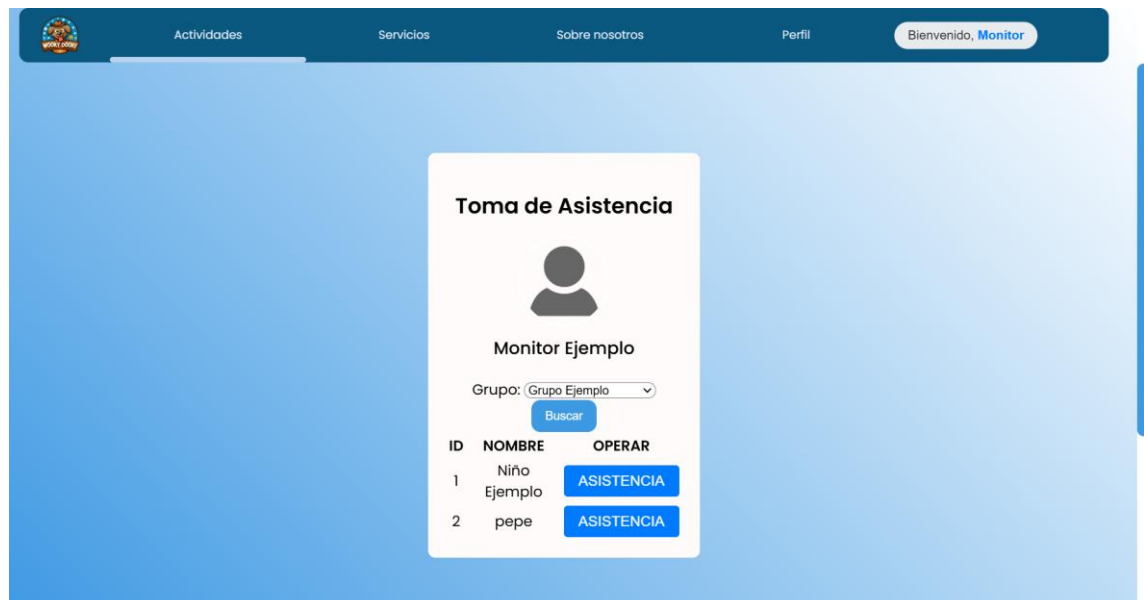
#### 1. Gestión del Perfil

Desde la sección de perfil, podrás modificar tu información personal, asegurándote de que los datos de contacto, foto de perfil y otra información relevante estén actualizados. Mantener esta información al día es importante para que los tutores y otros monitores puedan reconocerte fácilmente y comunicarse contigo cuando sea necesario.

#### 2. Pase de Lista de los Grupos Asignados

Uno de los aspectos clave de la plataforma es la posibilidad de llevar un control de asistencia de los niños que están bajo tu supervisión. Desde esta opción, podrás marcar la presencia o ausencia de los niños en cada actividad, facilitando el seguimiento y la seguridad dentro del centro.

Esta función también puede incluir observaciones sobre el comportamiento de los niños o cualquier incidencia que deba ser reportada a los tutores legales o a la administración del centro.

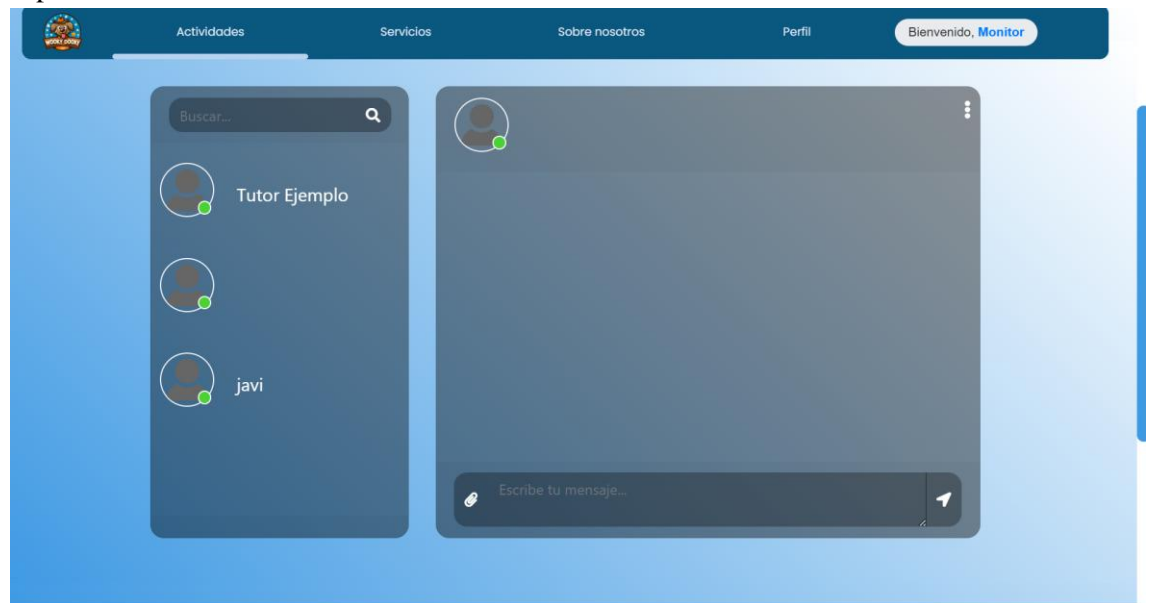


ID	NOMBRE	OPERAR
1	Niño Ejemplo	ASISTENCIA
2	pepe	ASISTENCIA

#### 3. Chat con Tutores Legales

Para mejorar la comunicación con los padres y tutores legales, la plataforma incluye una herramienta de mensajería directa. A través de este chat, podrás informar sobre el progreso de los niños, resolver dudas y mantener una comunicación fluida con las familias.

Este sistema de mensajería es especialmente útil para notificar cambios en horarios, resolver inquietudes o proporcionar detalles sobre el desarrollo de los niños en las actividades que supervisas.



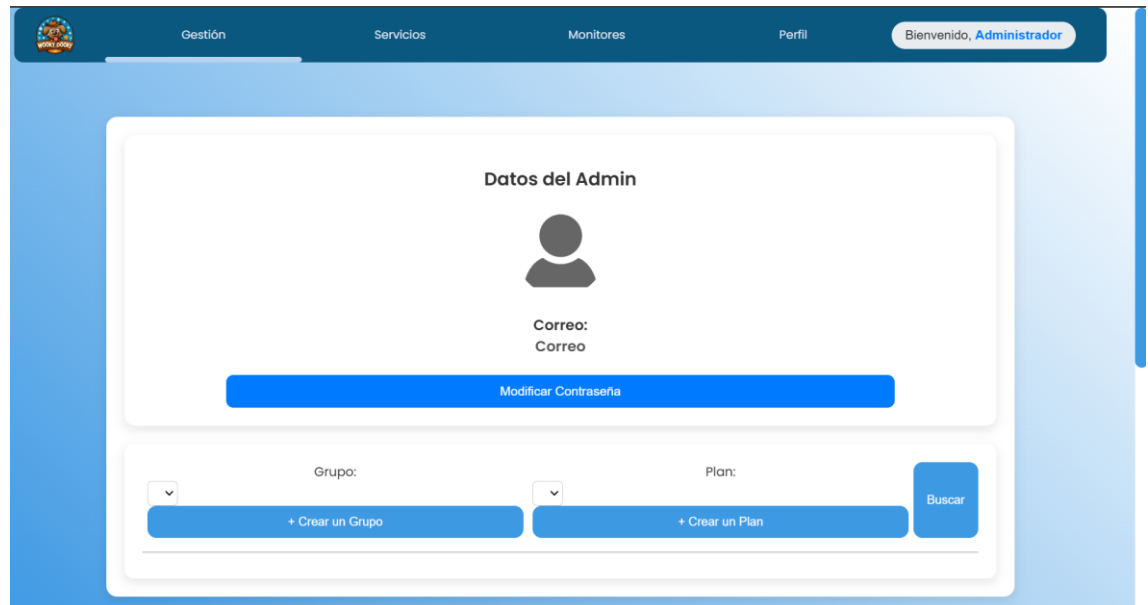
#### 4. Cierre de Sesión

Para garantizar la seguridad de tu cuenta y la privacidad de la información, es recomendable cerrar sesión una vez termines de utilizar la plataforma, especialmente si accedes desde un dispositivo compartido.



## Manual para Administradores - Guía de Uso y Funcionalidades

Tus responsabilidades incluyen la gestión de pagos, la inscripción de niños, la administración de monitores y la planificación de grupos y actividades. A continuación, se detallan las principales funcionalidades a las que tendrás acceso.



### Opciones para el Administrador

#### 1. Gestión de Pagos de los Padres

Como administrador, podrás supervisar y gestionar los pagos realizados por los tutores legales. Desde esta sección, tendrás acceso a la información de pagos, pudiendo verificar transacciones, procesar reembolsos si es necesario y asegurarte de que todas las cuotas han sido abonadas correctamente.

Esta función es clave para el correcto funcionamiento del campamento, ya que garantiza que todos los niños inscritos han completado el proceso de pago y pueden acceder a las actividades sin inconvenientes.

#### 2. Inscripción y Gestión de Niños

Desde la plataforma, podrás dar de alta a los niños inscritos en el campamento, asegurándote de que todos los datos personales, información médica y necesidades especiales estén correctamente registrados.

También podrás asignar a cada niño a un grupo específico según su edad, actividad o requerimientos específicos, facilitando la organización interna del campamento.

AVATAR	NOMBRE	PAGADO	ALERGIA	GRUPO	PLAN	ACCIONES
	Niño Ejemplo	1	Ninguna	Grupo Ejemplo (1)	null (null)	Operar Eliminar
	pepe	1	nada	Grupo Ejemplo (1)	algo (2)	Operar Ver su Actividad Eliminar

### 3. Creación y Gestión de Cuentas de Monitores

Como administrador, tendrás la posibilidad de gestionar el personal del campamento. Esto incluye:

Crear nuevas cuentas para monitores que se incorporen al equipo.

Asignarles grupos de niños y actividades específicas.

Modificar sus permisos dentro de la plataforma según sea necesario.

Eliminar cuentas de monitores que ya no formen parte del campamento.

Esta funcionalidad permite una gestión eficiente del equipo de trabajo, asegurando que cada monitor tenga acceso a la información y herramientas necesarias para desempeñar su labor.

Avatar	Nombre	Gmail	Definicion	Acciones
	Monitor Ejemplo	monitor@ejemplo.com	Monitor de ejemplo para el campamento.	Operar Eliminar Cambiar Contraseña

### 4. Creación de Nuevos Planes de Inscripción

Podrás diseñar y gestionar los diferentes planes y actividades a los que los niños podrán inscribirse. Desde esta sección, podrás:

Crear nuevos programas de actividades.

Establecer cupos de inscripción.

Definir requisitos o restricciones para cada plan.

Modificar o eliminar planes según la necesidad del campamento.

Esto permite mantener una oferta de actividades variada y bien organizada, asegurando que cada niño pueda participar en las opciones más adecuadas para su perfil.

Nombre Plan	Fechas	Inscripción	Precio	Definición	Acciones
	2025-06-01 - 2025-06-15	2025-05-25 (18:00:00)	100.00	Campamento de verano para niños de 6 a 12 años.	Operar VerNinos
algo	2025-03-02 - 2025-03-03	2025-03-02 (18:27:00)	500	algo	Operar VerNinos

## 5. Gestión de Grupos y Asignaciones

Para garantizar una mejor organización dentro del campamento, tendrás la opción de asignar niños y monitores a diferentes grupos según su edad, nivel de experiencia o tipo de actividad.

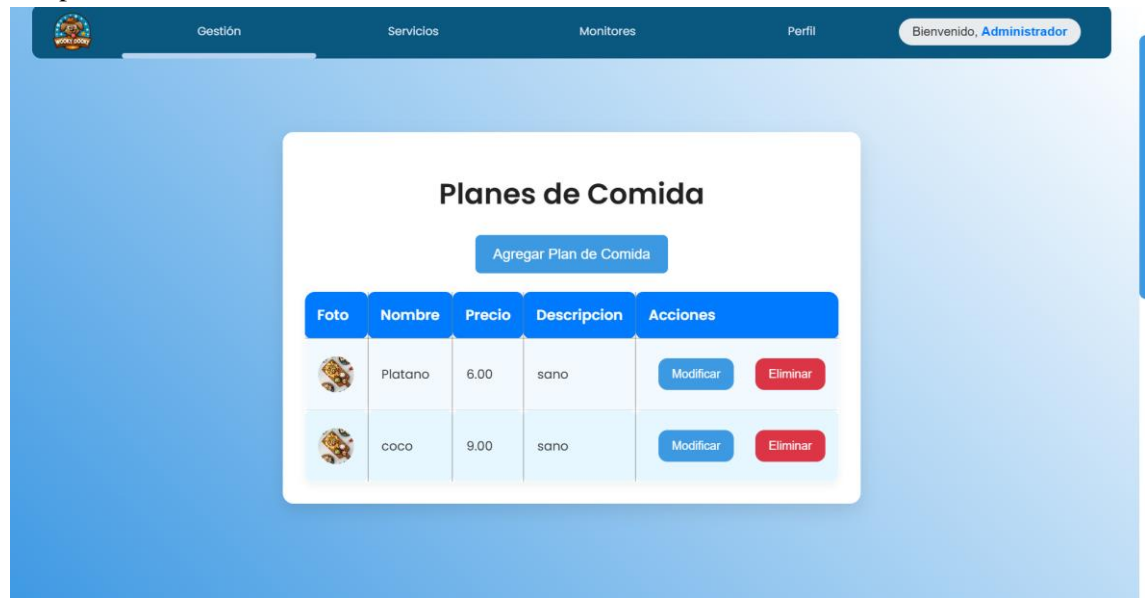
Esta función es esencial para mantener una estructura clara y efectiva, permitiendo una mejor distribución de responsabilidades y asegurando que cada niño esté en el grupo adecuado.

Nombre Grupo	Monito Principal	Operar
Grupo Ejemplo	Monitor Ejemplo	Operar VerNinos

## 6. Gestión de las Comidas del Campamento

El administrador también será responsable de organizar los

menús y las comidas que se ofrecerán durante el campamento.  
 Desde la plataforma, podrás:  
 Definir los menús semanales.  
 Adaptar las comidas a necesidades dietéticas especiales.  
 Gestionar proveedores y logística de alimentación.  
 Hacer ajustes en el menú según sea necesario.  
 Esto garantiza que todos los niños reciban alimentación adecuada y acorde a sus necesidades durante su estancia en el campamento.



## 7. Modificación de Datos del Usuario

Como administrador, podrás actualizar y modificar tu propia información de usuario dentro de la plataforma. Esto incluye datos personales, correo electrónico y contraseña, asegurando que siempre tengas acceso seguro a la gestión del campamento.

## 8. Cierre de Sesión

Para proteger la seguridad de la plataforma y la privacidad de los datos, es importante cerrar sesión al finalizar el uso, especialmente si accedes desde un dispositivo compartido.