

PROYECTO FINAL DE CICLO

**C.F.G.S ADMINISTRACIÓN DE
SISTEMAS INFORMÁTICOS EN RED**

**DISEÑO WEB Y ADMINISTRACIÓN DE BBDD
EN UN CENTRO JUVENIL**

Marco Batista Calado

Jerome Gamboa Cuison

Rubén Agyakwa Delgado

COLEGIO NUESTRA SEÑORA DEL PILAR

TÍTULO	Diseño web y administración de una BBDD de un centro juvenil.
AUTOR	Marco Batista Calado, Jerome Gamboa Cuisson y Rubén Agyakwa Delgado.
TUTOR	Begoña López
FECHA	13 de Junio del 2023
CICLO	Administración de Sistemas Informáticos en Red. [CFGs-ASIR]

RESUMEN DEL PROYECTO

El proyecto consiste en la creación de una página web para el Centro Juvenil La Balsa (Estrecho). La página web mostrará las actividades y ofertas del centro, permitirá la gestión de usuarios y la realización de formularios. Para atraer a más jóvenes, la página web incluirá contenidos visualmente atractivos como imágenes y videos de las actividades, así como información detallada sobre cada una de ellas.

Una característica importante de la página web será la capacidad de ofrecer formularios de registro para cada actividad. Estos formularios serán fáciles de usar y permitirán a los jóvenes inscribirse en línea y pagar por la actividad. Además, la página web incluirá una base de datos para mantener un registro de los participantes y las actividades.

ÍNDICE

I. DESCRIPCIÓN Y ALCANCE DEL PROYECTO	1
1. Descripción general del proyecto	1
2. Justificación.....	1
3. Objetivos	2
4. Destinatarios	2
5. Tecnología utilizada	3
II. PLANIFICACIÓN.....	4
1. Planificación inicial.....	4
AMBITOS.....	4
2. Planificación real.....	5
AMBITOS.....	5
III. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	6
1. Análisis de requerimientos	6
2. Diseño del sistema	6
a) Arquitectura Hardware.....	6
b) Arquitectura Software	7
Inspiraciones:	9
c) Restricciones (Hw y Sw).....	21
d) Aspectos de seguridad.....	21
e) Diseño del modelo de datos	29
f) Diseño de la interfaz de usuario (si procede)	31
IV. PLAN DE PRUEBAS	41
V. CONCLUSIONES	45
1. Objetivos cumplidos	45
2. Conclusiones	45
3. Futuras mejoras	46
VI. BIBLIOGRAFÍA	47

I. DESCRIPCIÓN Y ALCANCE DEL PROYECTO

1. Descripción general del proyecto

Consiste en la creación de una página web para el Centro Juvenil La Balsa (Estrecho) en la que se visualice las actividades y ofertas del centro, y permita la gestión de usuarios y de realización de formularios.

Para captar la atención de más jóvenes, la página web incluye una variedad de contenidos visualmente atractivos. Estos contenidos incluyen imágenes y videos de las actividades que se ofrecen, así como información detallada sobre cada una de las actividades. La información sobre cada actividad incluye detalles como la fecha, el lugar y el precio, así como una breve descripción de lo que se puede esperar durante la actividad.

Una de las características más importantes de la página web es su capacidad para ofrecer formularios de registro para cada una de las actividades. Estos formularios son fácilmente accesibles desde la página de cada actividad, y permiten a los jóvenes inscribirse en línea y pagar por la actividad en cuestión. Los formularios de registro están diseñados de manera clara y fácil de usar, lo que asegura que el proceso de registro sea rápido y sencillo para los usuarios.

Además de los contenidos y los formularios de registro, la página web también incluye una base de datos que permitirá al centro juvenil mantener un registro de los participantes y las actividades en las que se han registrado. Esta base de datos es fácilmente accesible por parte del personal del centro, lo que permite una mejor organización y gestión de las actividades.

2. Justificación

Este centro se dedica a ofrecer actividades y programas para jóvenes interesados en la música, los campamentos, la convivencia en grupo y sobre todo interesados en su relación con Dios, de forma que estos programas ayuden a los jóvenes a creer en su fe promoviendo los valores cristianos.

Realmente creemos que la razón principal de llevar a cabo este proyecto es aprovechar la oportunidad que brinda el TFG para realizar una acción de ayuda gratuita y desinteresada que realmente puede marcar la diferencia. La creación de esta herramienta de gestión es muy necesaria, ya que desde hace un par de años se ha convertido en una prioridad para el centro juvenil. Como animadores del centro, somos los más indicados para llevar a cabo este proyecto, que no termina aquí, sino que está diseñado para mejorar y evolucionar en el futuro.

3. Objetivos

El objetivo principal de la página web es ofrecer una plataforma atractiva y fácil de usar para que los jóvenes puedan ver la oferta de actividades del centro y registrarse para participar en ellas. Por ello la página web incluye una amplia gama de contenidos, así como formularios de registro y una base de datos para mantener un registro de los participantes y las actividades. Y otro de los objetivos principales es que el centro juvenil salesiano podrá llegar a un público más amplio y ofrecer sus actividades de manera más eficiente y efectiva, además de que facilite también las tareas de administración para el director del centro.

4. Destinatarios

Los destinatarios de este proyecto son todos aquellos involucrados en el Centro Juvenil de Estrecho, incluyendo a los animadores, los directores de la Obra Salesiana, los jóvenes de entre 9 y 18 años que participan en el centro y, sobre todo, sus familias. La creación de esta herramienta de gestión y registro de actividades no solo beneficiará a los participantes actuales del centro, sino que también puede ser adaptada y utilizada en otros centros juveniles en función de sus necesidades específicas.

Este proyecto está diseñado para abarcar a una amplia audiencia y ofrecer beneficios tangibles para todos los involucrados en la comunidad del centro juvenil. Desde los animadores y directores que necesitan herramientas efectivas para la gestión de las actividades, hasta los jóvenes y sus familias que desean tener acceso fácil y rápido a la información sobre las actividades y eventos del centro, este proyecto ofrece una solución integral y práctica. Además, la escalabilidad del proyecto lo hace ideal para ser adaptado y utilizado en otros centros juveniles, ayudando a satisfacer las necesidades de las comunidades juveniles en todo el mundo.

5. Tecnología utilizada

El proyecto se desarrolla utilizando una variedad de tecnologías modernas, como **HTML**, **CSS**, **JavaScript** y **PHP**. Estas herramientas son esenciales para crear una página online de calidad, eficiente y que sea fácil de navegar.

Para el entorno de programación utilizamos el programa como **Visual Studio Code**, un software de código abierto que es muy efectivo para crear aplicaciones online de alta calidad. En conjunto con variedad de plugins como PHP Debug, Microsoft Edge Tools o HTML CSS Support.

Se utilizó una máquina virtual con el **sistema operativo Xubuntu 16.04** instalado y el **servidor web Apache 2.4.53**. Xubuntu es una distribución de Linux basada en Ubuntu que se caracteriza por ser ligera y rápida, lo que la convierte en una opción ideal para la creación de una máquina virtual para alojar un servidor web. Por otro lado, Apache es uno de los servidores web más populares y confiables del mundo, utilizado por miles de sitios web en todo el mundo.

Al utilizar esta combinación de tecnologías, se garantiza un alto nivel de rendimiento y seguridad para la página web del centro juvenil. Además, esta configuración es altamente escalable y fácil de mantener, lo que la hace ideal para el desarrollo de futuras mejoras y actualizaciones en el futuro.

En el ámbito de la comunicación, hemos utilizado **Discord** y **Anydesk** para facilitar la interacción. Discord nos ha permitido compartir pantallas y comunicarnos, mientras que Anydesk ha sido útil para acceder a las máquinas y brindarnos ayuda mutua. Además, hemos utilizado Drive como plataforma de almacenamiento para mantener actualizadas las mejoras de la página.

Finalmente, para alojar nuestro servicio en la nube, hemos utilizado una combinación de **DDNS** (Dynamic DNS) y el **servicio no-IP**. Además, hemos realizado las configuraciones necesarias en nuestro router para redirigir el tráfico entrante al servidor alojado en la nube. Esto implica establecer reglas de reenvío de puertos y asegurarnos de que el firewall del router permita el acceso al servicio. Estas configuraciones nos han permitido superar las restricciones de red y asegurar una conectividad confiable entre el cliente y el servidor, sin importar la ubicación física.

II. PLANIFICACIÓN

1. Planificación inicial

La planificación inicial de nuestro proyecto se centrará en hablar de los momentos libres en los que podremos conectarnos y trabajar en equipo. Reconocemos la importancia de establecer horarios de trabajo en equipo que nos permitan colaborar de manera efectiva y asegurarnos de avanzar de manera coordinada hacia nuestros objetivos. Muy importante también la evaluación de lo realizado y los ajustes, es decir, revisaremos periódicamente nuestra planificación inicial (que vamos a hacer esa semana) y evaluaremos su efectividad. Si identificamos dificultades o conflictos recurrentes con los horarios establecidos, estaremos abiertos a realizar ajustes y buscar soluciones alternativas que satisfagan las necesidades de todos los miembros del equipo.

Rubén y Marco serán responsables principalmente del desarrollo en PHP, mientras que Jerome se enfocará en HTML/CSS. Sin embargo, todos estaremos presentes para aprender y brindar ayuda conjuntamente. En cuanto a la base de datos, se trabajará de forma colaborativa con el objetivo de generar la mayor cantidad de ideas posibles para el almacenamiento de datos. A continuación dejamos una aproximación de horas que van a ser empleadas en cada en diferentes ámbitos:

AMBITOS	HORAS
Diseño de tablas	6
Programación registro y addhijo	12
Programación login	2
Programación logout	2
Programación perfil	2
Programación newpassword	2
Programación usuario	4
Programación misreservas	4
Programación formularios	10
Programación funciones	50
Programación HTML/CSS	100
Documentación	40
TOTAL = 234	

2. Planificación real

La planificación real de nuestro proyecto se ha basado en la necesidad de adaptar nuestros horarios y tiempos libres debido a las dificultades (dudas, problemas de código, mucha investigación) que hemos encontrado, las cuales han retrasado nuestro trabajo. Hemos reconocido la importancia de ajustar nuestros horarios de trabajo en equipo para poder cumplir con una entrega adecuada del proyecto. Además, hemos dado especial énfasis a la evaluación continua de nuestro progreso y la disposición a realizar ajustes cuando sea necesario.

Revisión y Análisis de Dificultades: Tras encontrarnos con dificultades que impactaban nuestro avance, hemos llevado a cabo una revisión exhaustiva de la situación. Identificamos los obstáculos específicos que enfrentamos y evaluamos su impacto en nuestros horarios y tiempos libres. Esta revisión nos permitió comprender las limitaciones y desafíos que estábamos enfrentando y nos motivó a buscar soluciones efectivas.

Adaptación de Horarios: Ante las dificultades, realizamos ajustes en nuestros horarios individuales y de trabajo en equipo. Dedicamos tiempo extra fuera de nuestros momentos libres para abordar tareas pendientes y recuperar el tiempo perdido. Además, flexibilizamos nuestra disponibilidad durante los días laborables para colaborar de manera más eficiente.

Aproximadamente está es la cantidad de horas realizadas en diferentes ámbitos del proyecto:

AMBITOS	HORAS
Diseño de tablas	21
Programación registro y addhijo	21
Programación login	6
Programación logout	2
Programación perfil	4
Programación newpassword	6
Programación usuario	6
Programación misreservas	6
Programación formularios	24
Programación funciones	84
Programación HTML/CSS	115
Documentación	46
TOTAL = 341	

III. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

1. Análisis de requerimientos

1. **Navegadores:** Nuestra página web está diseñada para brindar una experiencia óptima en diferentes navegadores. Entre las opciones recomendadas, destacamos tres navegadores que se adecuan especialmente a nuestras características y funcionalidades: **Google Chrome, Firefox y Microsoft Edge.**
2. **Requerimientos de hardware del servidor:** hemos utilizado una máquina virtual de 64 bits con las siguientes especificaciones: 3240MB de memoria RAM, 1 CPU, 12MB de memoria de video y un almacenamiento virtual que actualmente ocupa aproximadamente 17GB. Estas configuraciones nos permiten garantizar un rendimiento óptimo y una capacidad suficiente para satisfacer las necesidades de nuestro servidor.

2. Diseño del sistema

a) Arquitectura Hardware

La arquitectura de hardware utilizada en nuestro proyecto se compone de varios elementos clave que interactúan para ofrecer una solución web robusta y confiable. A continuación, describiremos cada uno de estos componentes y su papel en la arquitectura general:

1. **Servidor Apache:** El servidor Apache es el componente central de nuestra arquitectura. Se ejecuta en la máquina virtual con sistema operativo Xubuntu 16.04 y es responsable de gestionar las solicitudes web entrantes y entregar las páginas web solicitadas a los clientes. El servidor Apache es conocido por su estabilidad, escalabilidad y soporte para diferentes tecnologías web.
2. **PHPmyadmin:** su integración con el servidor Apache es una buena opción para el proyecto por varias razones. En primer lugar, es una herramienta de administración de bases de datos ampliamente utilizada y confiable. Proporciona una interfaz gráfica intuitiva que facilita la gestión de la base de datos, lo que resulta en una mayor eficiencia y productividad.

Además, PHPMyAdmin es compatible con varios sistemas de gestión de bases de datos, incluyendo MySQL, MariaDB y otros, lo que te brinda flexibilidad para elegir el sistema que mejor se adapte a tus necesidades.

Esta herramienta también ofrece numerosas funciones y características útiles, como la capacidad de crear, modificar y eliminar tablas, ejecutar consultas SQL, importar y exportar datos, realizar copias de seguridad y restauraciones, y gestionar usuarios y permisos de acceso.

Otra ventaja de PHPMyAdmin es que es de código abierto, lo que significa que es gratuito y tiene una comunidad activa de desarrolladores que lo mantienen actualizado y mejoran constantemente.

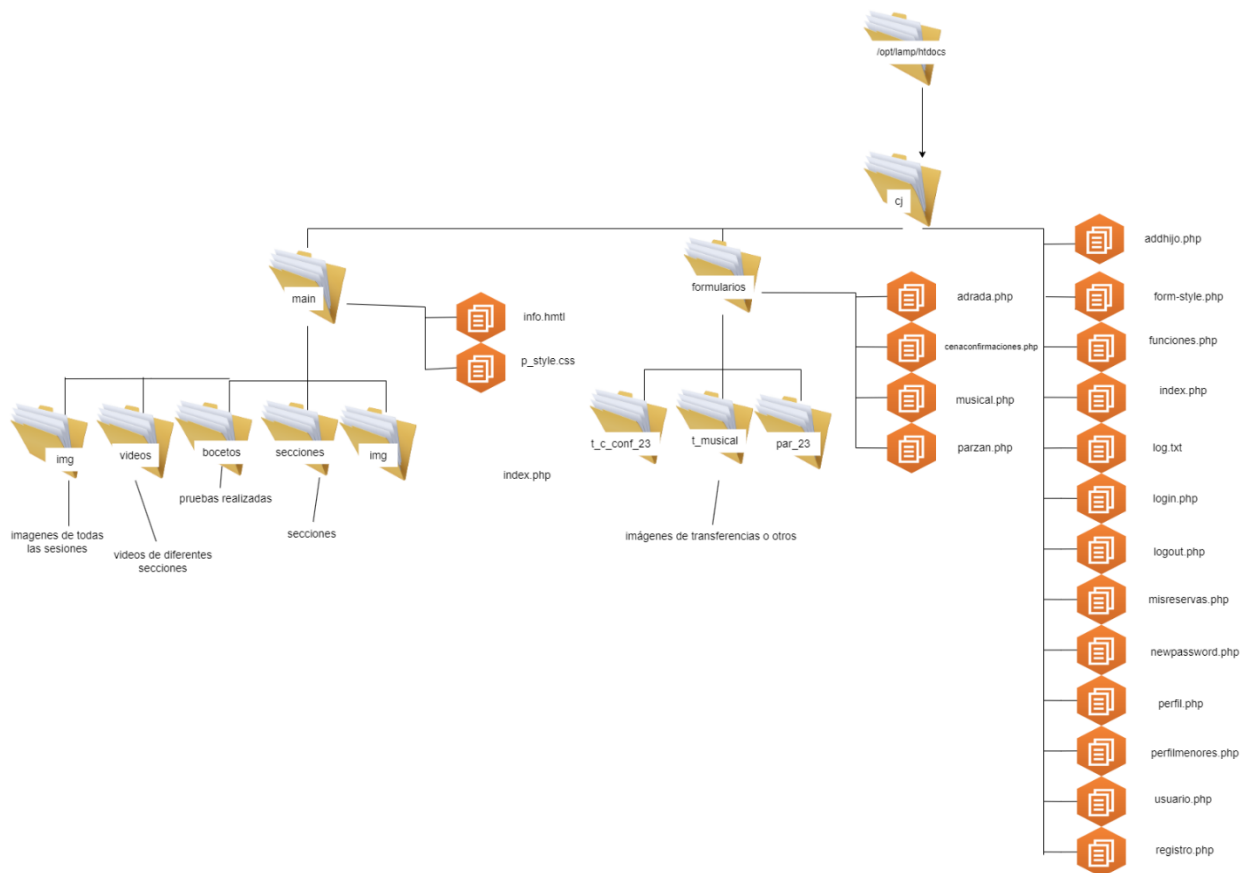
Posteriormente comentamos la modificación de algunas directivas de archivos de configuración.

3. **Máquina virtual:** Hemos optado por utilizar una máquina virtual con el sistema operativo Xubuntu 16.04 para alojar nuestro servidor Apache. La máquina virtual nos permite crear un entorno de desarrollo y producción aislado, lo que facilita la configuración, administración y mantenimiento del servidor web. Además, al utilizar el modo adaptador puente, la máquina virtual obtiene una dirección IP fija y se integra en la red de manera similar a una máquina física.
4. **Router:** El router desempeña un papel esencial en nuestra arquitectura, ya que actúa como un enlace entre el servidor web y la red externa, como Internet. Permite la comunicación bidireccional y el enrutamiento de los datos entre el servidor y los clientes. Además, hemos realizado configuraciones específicas en el router para redirigir el tráfico entrante hacia el servidor alojado en la máquina virtual. Esto implica establecer reglas de reenvío de puertos y asegurarnos de que el firewall del router permita el acceso al servicio.

b) Arquitectura Software

La página web es el resultado final del desarrollo web utilizando HTML, CSS, JavaScript y PHP. La página web se almacena en el servidor Apache y se entrega a los clientes cuando solicitan acceder al sitio. La arquitectura del servidor Apache y el sistema operativo Xubuntu proporcionan un entorno seguro y confiable para alojar la página web. Antes de profundizar en los aspectos clave del código, es importante tener una visión general de la estructura de carpetas y archivos que hemos utilizado en nuestro proyecto.

A continuación, presentamos una imagen que muestra la estructura de carpetas y archivos:



Inicialmente, esta estructura no era la planificada, pero debido a imprevistos y dificultades en la integración de PHP y HTML, nos vimos obligados a adaptarla en el último momento. A pesar de esto, hemos encontrado que esta estructura nos ha facilitado el trabajo y ha permitido una gestión eficiente del código fuente y los recursos del proyecto. A continuación, destacaremos algunos aspectos relevantes del código que hemos investigado y desarrollado:

HTML/CSS

Principalmente enfatizamos que hemos desarrollado la página web con **codificación manual**, escribiendo cada línea de código nosotros mismos. Esto ha requerido una gran investigación y actualización constante de nuestros conocimientos, lo que nos ha hecho más conscientes y comprometidos con cada aspecto del proceso. Al realizar el código desde cero, hemos tenido un mayor control sobre la funcionalidad y personalización de la página, permitiéndonos adaptarla de manera precisa a nuestras necesidades y ofrecer una experiencia única a los usuarios. Aunque esta metodología de codificación manual ha traído consigo desafíos, especialmente en términos de conocimientos y trabajo en equipo, ya que no es lo mismo realizar los cambios de forma individual que trabajar en colaboración con varias personas. La falta de ciertos conocimientos y la necesidad de mantenernos al día con las mejores prácticas nos ha exigido un esfuerzo adicional para superar obstáculos y garantizar la coherencia y eficiencia del proyecto.

Inspiraciones:

En primer lugar investigamos qué elementos podríamos agregar a nuestra página web, buscando bocetos que se ajustaran a nuestros criterios. Nos centramos principalmente en:

[Salesianos Estrecho](#)

[Salesianos Atocha](#)

[Centro Juvenil Salesianos de Atocha](#)

Después de eso, realizamos un análisis para identificar las etiquetas que no habíamos utilizado a lo largo del curso, con el objetivo de facilitar la distribución de la página. Esto nos permitiría realizar cambios de manera más sencilla y cómoda. Utilizamos HTML5, que es el lenguaje de programación que empleamos para estructurar nuestras páginas web.

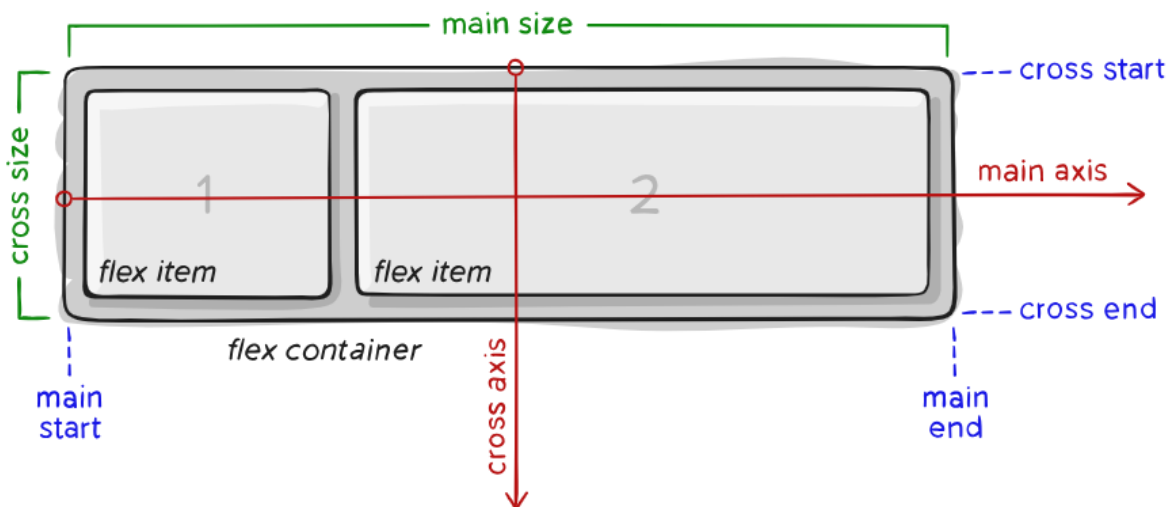
Además, aprendimos cómo agregar un **favicon** a la página mediante el siguiente código:

```
<link rel="icon" type="image/x-icon" href="Enlace de donde este el icono">
```

También exploramos diversas funcionalidades de CSS y tuvimos que importar varios recursos que encontramos durante nuestra investigación.

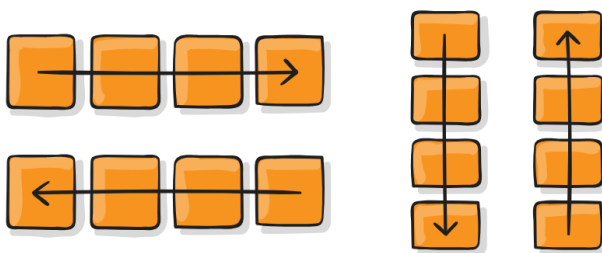
Uno de los conceptos en los que nos hemos apoyado ampliamente es **display: flexbox**. Esta propiedad CSS nos ha permitido adaptar nuestro menú de forma responsiva en primer lugar. Sin embargo, posteriormente fue necesario utilizar una **media query** para crear una versión específica para dispositivos móviles.

Flexbox es una propiedad del apartado '**display**' que proporciona una forma más cómoda de organizar y alinear elementos dentro de un contenedor. Con Flexbox, podemos lograr disposiciones flexibles y adaptativas, lo que facilita el diseño de la estructura de una página web de manera más eficiente.



Nos permite colocar los elementos que tenga la caja de diferentes maneras con el **Flex-direction:**

Flex-direction



Permitiéndonos colocarlo ya sea en columnas o en filas en reversa o no.

```
.nombre_caja {
  flex-direction: row | row-reverse |
  column | column-reverse;
}
```

```
.c_tarjetas {
  width: 100%;
  margin: auto;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}
```

Últimas noticias



Campamento Parzán 2023

¿Qué aventuras y desafíos aguardan en el campamento PARZÁN 2023? Únete a este misterioso evento organizado por los centros juveniles salesianos y descubre cómo la integración, la salud y la superación se

Un ejemplo de cómo se vería, lo que hace es que las tarjetas se vean en forma de columna.

Justify-content:

Define el cómo se van alinear los ítems que están dentro de la caja en el que estén.

- **flex-start:** los ítems se alinean a lo largo del borde de inicio.
- **flex-end:** los ítems se alinean a lo largo del borde final.
- **center:** los ítems están centrados en la línea.
- **space-between:** los ítems se distribuyen uniformemente a lo largo de la línea.
- **space-around:** los ítems se distribuyen en línea con el mismo espacio entre ellos.
- **space-evenly:** los ítems se distribuyen de manera que el espacio entre dos elementos cualesquiera en la línea.

flex-start



flex-end



center



space-between



space-around



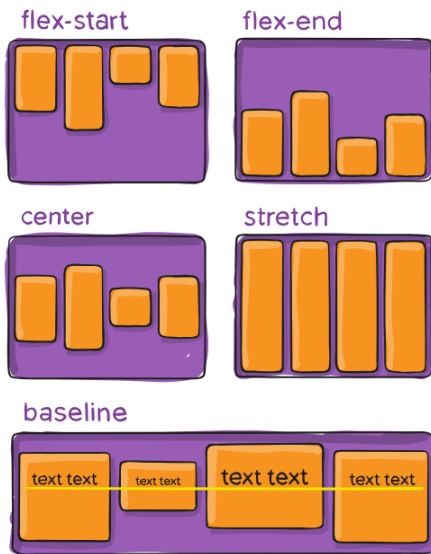
space-evenly



```
flex-container {  
justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;  
}
```

Align-items

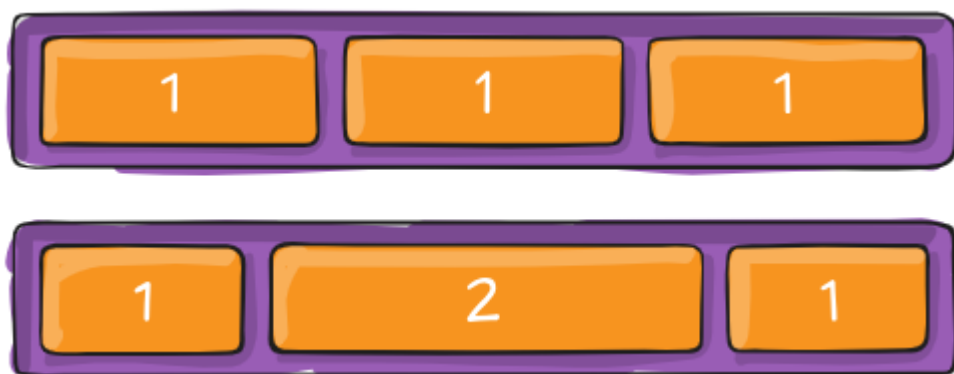
Establece el comportamiento de los items alineados según su eje transversal (*main axis*).



Aquí también podemos ver que el contenido queremos que nos lo alinee de una forma concreta.

```
.c_tarjetas {
  width: 100%;
  margin: auto;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}
```

Flex-grow



Define la capacidad para que un ítem crezca ej: si el flex-grow fuese de 1 para todos los ítems entonces el ítem que tiene 2 en el flex-grow será el doble que las que tiene 1. (no aparece en la página pero se está pensando en añadirlo para algunas partes).

Hemos utilizado la etiqueta **!important** para solucionar problemas específicos en nuestro diseño. En particular, nos ayudó a corregir un problema en el que se generaba un espacio en blanco no deseado en el lado derecho de la página. Al aplicar **!important** a ciertas reglas de estilo, pudimos asegurarnos de que esas reglas prevalecieran sobre otras y resolvieran el problema de manera efectiva. Esta etiqueta es útil cuando necesitamos dar prioridad a ciertas reglas de estilo sobre otras.

```
▼ html, body{  
  max-width: 100% !important;  
  overflow-x: hidden !important;  
  width: max-content;  
}
```

Otra etiqueta que hemos usado bastante ha sido la de **visibility** que se trata de una propiedad que indica si el item en el que está colocada queremos que se vea o no, se divide en visibility, hidden o collapse. Las que más se suelen usar son la de “visibility” o “hidden”, que son para que se vean o no respectivamente.

```
#active {  
  visibility: hidden;  
}  
  
.fas {  
  visibility: hidden;  
}
```

Con esto logramos que se oculte los botones para el menú en la versión de móvil.



Los filtros que usamos para ver de forma más atractiva imágenes han sido: *mix-blend-mode* y *filter*. El primero es principalmente para realizar mezclas con el contenido y el fondo y el *filter* era para darle propiedades a la imagen si brillo, opacidad, contraste entre otros...


```
.imgrrss {  
  height: auto;  
  mix-blend-mode: luminosity;  
  filter: brightness(100%);  
}
```

Para asegurarnos que las imágenes que pongamos se pongan de color blanco.

Hemos utilizado la propiedad “transform” de manera destacada en la creación de la portada de nuestra página web.

```
.c_cont-portada {  
  margin-top: 10vh;  
  width: 100%;  
  max-height: 600px;  
  height: 100vh;  
  transform-style: preserve-3d;  
  display: flex;  
  justify-content: center;  
  flex-direction: column;  
  align-items: center;  
}
```



Al aplicar esta propiedad al elemento correspondiente, hemos logrado dotarlo de un efecto tridimensional. Esta propiedad cuenta con dos estados el primero es el estado “flat”, que representa su configuración predeterminada en dos dimensiones (2D), y el segundo es el estado “preserve-3d”, que le otorga el estilo tridimensional deseado.

Una vez puesto en 3d ya podemos realizar desplazamientos con este elemento, de manera que veamos las imágenes desplazadas en un lado o en otro, para ello volveremos a usar la propiedad de *transform* pero esta vez usaremos.

- **Tranlatex/y(x):** esta propiedad indicará cuánto queremos que se se desplace en el eje x o y nuestro elemento.
- **Scale:** esta propiedad indicará cuánto queremos que se reduzca o se amplíe en función de nuestro gusto, en nuestro caso fue reducirlo.

Parecido a los `background` de los que hemos hablado anteriormente también está la propiedad del `"background-size"` para las `` `<video>` con la propiedad `object-fit` que tiene las características: `fill/ contain/ cover/...`

Para crear el índice recortado, nos inspiramos en una plantilla que descubrimos, la cual utiliza una propiedad que no habíamos explorado en clase: `"clip-path"`. Esta propiedad permite realizar un recorte preciso, ocultando la región externa al área recortada. En nuestro caso, decidimos utilizar una forma circular, por lo que utilizamos la función `"circle"` para definir el tamaño y la ubicación del recorte. Para lograr esta ubicación precisa, empleamos la función `"calc"`, que, aunque aún me resulta un concepto algo complejo, básicamente nos permite realizar operaciones entre píxeles y porcentajes.

```
.wrapper {  
  position: fixed;  
  z-index: 90;  
  top: 0;  
  left: 0;  
  height: 100%;  
  width: 100%;  
  /*background: linear-gradient(-135deg, #c850c0, #4158d0);*/  
  /* background: linear-gradient(375deg, #1cc7d0, #2ede98); */  
  /* background: linear-gradient(-45deg, #e3eefe 0%, #efddfb 100%);*/  
  /* clip-path: circle(25px at calc(0% + 45px) 45px); */  
  background: #db1b13;  
  clip-path: circle(25px at calc(100% - 45px) 45px);  
  transition: all 0.3s ease-in-out;  
  overflow-y: auto;  
}
```

Además, encontramos una sección que finalmente decidimos no modificar, pero es importante mencionarla: el uso de `"cursor:pointer"`. Esta propiedad define cómo queremos que se vea el cursor al pasar sobre el elemento al que se aplica.

Por último, cabe destacar el uso de los elementos "checked" en la sección de la portada. Utilizamos botones para cambiar las fotos, y cuando uno de ellos está marcado como "checked", se aplican las propiedades CSS correspondientes a ese elemento, lo que permite que se muestre de manera efectiva en la página.

Se ha realizado un cuidadoso diseño que permite la visualización en dos formatos distintos: la versión de escritorio y la versión móvil. Para lograr esto, se implementó un mediaquery de 600px, asegurándonos así de abarcar la gran mayoría de los dispositivos móviles disponibles en el mercado. En cuanto al formato de tablet, se tomó la decisión de omitirlo debido a que no todos los usuarios hacen uso de este tipo de dispositivos. Sin embargo, es importante mencionar que se considera como una posible ampliación del programa en el futuro, en caso de que la demanda o necesidad así lo requieran. Esta decisión busca optimizar la experiencia de usuario y garantizar un despliegue eficiente en los dispositivos más utilizados en la actualidad.

Adjuntamos vídeo de la navegación web en un dispositivo Iphone 12 Pro y navegador Safari:

[Responsive móvil](#)

PHP

Desde el inicio de nuestro proyecto, tomamos la decisión de utilizar ficheros PHP únicos que contengan todos los formularios correspondientes a cada una de nuestras necesidades. En aquel entonces, con nuestro nivel de conocimiento limitado, creíamos que esta era la única forma de mostrar de manera sencilla los errores de los campos del cliente utilizando JavaScript.

Esta elección se basó en la idea de tener un enfoque centralizado y organizado para el manejo de formularios. Al combinar todos los formularios relacionados en un solo fichero PHP, teníamos la ventaja de mantener un control más fácil sobre la lógica de validación y procesamiento de datos.

Además, al tener todos los formularios en un solo lugar, pudimos implementar funcionalidades compartidas y reutilizar código de manera eficiente. Esto nos permitió evitar la duplicación innecesaria de código y facilitó el mantenimiento y actualización futura de nuestros formularios.

Asimismo, la elección de utilizar JavaScript para mostrar los errores de los campos del cliente fue una decisión acertada en ese momento. Esta tecnología nos permitió realizar validaciones en tiempo real, brindando retroalimentación inmediata al usuario sobre posibles errores en sus datos. De esta manera, pudimos mejorar la experiencia del usuario y reducir la posibilidad de envío de datos incorrectos.

Sin embargo, a medida que nuestro conocimiento y experiencia han ido creciendo, nos hemos dado cuenta de que existen otras formas de abordar la estructura y gestión de formularios en proyectos más complejos. El uso de enfoques como la separación de responsabilidades y la arquitectura de software modular pueden proporcionar beneficios adicionales en términos de mantenibilidad, escalabilidad y legibilidad del código.

Hablar de reutilización eficiente de código nos lleva a destacar la metodología de funciones empleando la tecnología **"include" y "require"**. Esta práctica es crucial para optimizar el desarrollo de software y garantizar una gestión eficiente del código.

Cuando hablamos de "include" y "require", nos referimos a directivas en lenguajes de programación como PHP, que nos permiten importar y reutilizar fragmentos de código en diferentes partes de un programa. Estas directivas facilitan el uso de funciones o clases definidas en otros archivos, lo que ahorra tiempo y esfuerzo en la creación de código desde cero.

La ventaja principal de utilizar "include" y "require" radica en la modularidad. Al dividir el código en fragmentos lógicos separados, podemos mantener un enfoque más estructurado y legible.

Además, si hay cambios o mejoras que deban aplicarse a una función específica, solo es necesario modificar un único archivo en lugar de buscar y editar múltiples secciones dispersas por todo el código.

Otra ventaja significativa es la reutilización de código. Al emplear funciones incluidas o requeridas en diferentes partes de un programa, evitamos la repetición innecesaria de código y promovemos la consistencia en el comportamiento de nuestras aplicaciones.

Sin embargo, es importante tener en cuenta que **la elección adecuada entre "include" y "require" dependerá de la situación particular**. "Include" nos permite continuar la ejecución del programa incluso si el archivo incluido no se encuentra, mientras que "require" genera un error fatal en caso de no encontrar el archivo necesario. Esto implica que "require" puede ser más apropiado cuando la inclusión del archivo es crítica para el funcionamiento del programa.

En nuestro caso creemos que la elección adecuada sin duda es require, por la estructura de proyecto que tenemos.

```
// Requerir las funciones del fichero funciones.php para utilizarlas.  
require 'funciones.php';
```

registro.php: este archivo merece especial atención debido a su importancia crucial para nuestra base de datos y proyecto. Este archivo se encarga de realizar un registro completo, lo cual es fundamental para nuestro sistema. Además, se complementa con el archivo **"addhijo.php"**, un fichero en el que se ha prestado especial atención a las consultas adecuadas. Se han realizado diversas verificaciones, como por ejemplo: comprobar si el usuario marcó la opción de tener hijos al registrarse (representada como un valor booleano "1"), y si posteriormente ingresó un número de hijos (por ejemplo, "2"), recopilados en el campo "tipo" de la tabla "usuario" y en la tabla "adulto" bajo la columna "hijos". Además, se realiza una comparación entre el número de hijos especificados y el número de hijos ya registrados en la tabla "menores", para evitar agregar más de los permitidos.

- Entre tantas validaciones resaltar parte de código de la función **validarDNI()**:

```
} else {  
$numeros = substr($dni, 0, 8);  
$letra = strtoupper(substr($dni, -1));  
$letra_calculada = substr('TRWAGMYFPDXBNJZSQVHLCKE', $numeros % 23, 1);  
if ($letra != $letra_calculada) {  
$error_dni = "La letra no coincide";  
$contador1++;  
} else {
```

Si el DNI pasa las validaciones anteriores, se procede a verificar que la letra coincida con el número. Para ello, se obtienen los primeros 8 caracteres (números) y la última letra del DNI.

Luego, se calcula la letra esperada utilizando una fórmula basada en el número obtenido. Si la letra calculada no coincide con la letra del DNI, se establece un mensaje de error y se incrementa el contador.

Además, esta función también realiza la importante tarea de verificar la existencia del DNI introducido en las tablas adultos y menores de nuestra base de datos, evitando así duplicados.

login.php:

Las **sesiones** son una forma de mantener la información del usuario en un sitio web a lo largo de varias solicitudes. Sirven para almacenar datos específicos del usuario, como preferencias, estado de autenticación o carrito de compras, durante su interacción con el sitio.

En este código, se utiliza el concepto de sesiones para gestionar la autenticación de los usuarios. Al iniciar sesión correctamente, se crea una sesión para el usuario utilizando el siguiente código: `$_SESSION['usuario'] = $usuario;`

Esta línea de código guarda el nombre de usuario en una variable de sesión llamada 'usuario'. Esta información se puede utilizar en otras páginas del sitio para verificar si el usuario ha iniciado sesión y personalizar la experiencia del usuario en función de su estado de autenticación. Para ello se tiene que abrir siempre la sesión y recoger el dato, pero antes de recoger el dato, como en todos los ficheros de nuestro proyecto nos interesa que para acceder a ellos haya una sesión activa, creamos la función **verificarSesion()** que simplemente comprueba que hay una sesión activa y si no la hay, redirir al cliente a la página principal:

Código a implementar en cada fichero antes de la etiqueta html

```
session_start();  
//Verificar que se ha iniciado sesión y si no redirigirle.  
verificarSesion();  
$usuario=$_SESSION['usuario'];
```

Función verificarSesion():

```
function verificarSesion() {  
    // Verificar si el usuario ha iniciado sesión  
    if (!isset($_SESSION['usuario'])) {  
        // Si el usuario no ha iniciado sesión, redirigirlo a la página de inicio de sesión  
        header('Location: /cj/login.php');  
        exit();  
    }  
}
```

Las sesiones también se utilizan para establecer una cookie de sesión que almacena el nombre de usuario durante un día (86400 segundos): `setcookie('nombre_usuario', $usuario, time() + 86400, '/');` Esta cookie se envía al cliente y se utiliza para identificar al usuario en futuras solicitudes, evitando que tengan que volver a ingresar su nombre de usuario.

logout.php:

```
<?php
session_start(); // Iniciar sesión
if(isset($_SESSION['usuario'])){ // Verificar si la variable de sesión 'usuario' está definida
session_unset(); // Eliminar todas las variables de sesión
session_destroy(); // Destruir la sesión actual
}
header("Location: login.php"); // Redirigir al usuario a la página de inicio de sesión
exit(); // Detener la ejecución del script
?>
```

newpassword.php: muy interesante saber esta forma de inicializar las variables:

```
// Inicializar las variables de error
$currentPasswordErr = $newPasswordErr = $confirmPasswordErr = $confirmPasswordErr2 =
'';
```

musical.php: uno de los formularios como ejemplo de su arquitectura en general y con la muestra de errores.

```
<label for="num_entradas">Número de entradas:</label>
<input type="number" id="num_entradas" name="num_entradas" min="1" max="6"
onfocus="quitarError('error_num_entradas')" required>
<?php if(isset($error_num_entradas)) { ?>
<span id="error_num_entradas" style="color:red;"><?php echo $error_num_entradas;
?></span><br>
<?php } ?>
```

Otra función de vital importancia en nuestro proyecto es **guardarErrorEnLog(\$error)**. Esta función se encarga de registrar todos los errores que se producen al realizar cualquier operación en la base de datos. Es llamada en cada bloque **catch** de nuestro código, permitiéndonos capturar y almacenar de manera adecuada cualquier error que pueda ocurrir durante la ejecución del programa. Esto resulta fundamental para tener un seguimiento detallado de los errores y facilitar su posterior análisis y corrección. En el bloque catch además se llama a una variable global declarada también en el fichero funciones que muestra al usuario un mensaje de error: “Se ha producido un error. Habla con el administrador.”

```
//Función guardar errores en un fichero log.
function guardarErrorEnLog($error) {
    $fecha = date('Y-m-d H:i:s');
    $mensaje = "[".$fecha.$error". PHP_EOL;
    $archivo = 'log.txt';
    // Intentar abrir el archivo en modo 'append'
    if ($handle = fopen($archivo, 'a')) {
        fwrite($handle, $mensaje); // Escribir en el archivo
        fclose($handle); // Cerrar el archivo
    }
}
```

Ejemplo catch:

```
} catch (Exception $e) {
    $error = $e->getMessage();
    guardarErrorEnLog($error);
    global $habladmin;
    die (" $habladmin");
}
```

Por último, hablemos sobre el **método de pago** utilizado para actividades y formularios. En este caso, seguimos utilizando el método tradicional que ha estado en uso en nuestro centro durante años, que consiste en adjuntar una **foto de la transferencia realizada**. Aunque hemos investigado la posibilidad de implementar un **Terminal de Punto de Venta** (TPV) virtual de forma gratuita, lamentablemente está fuera de nuestras posibilidades en este momento. Por lo tanto, continuaremos utilizando el método de adjuntar una foto de la transferencia como forma de pago, asegurándonos de mantener la práctica existente en nuestro centro.

c) Restricciones (Hw y Sw)

Hemos enfrentado algunas restricciones importantes en nuestro proyecto. En primer lugar, el servidor utilizado no está preparado para manejar una alta demanda, ya que tiene limitaciones de hardware. En cuanto al software, construir la página web desde cero ha implicado ciertas limitaciones. Hubiéramos tenido más flexibilidad y funcionalidad si hubiéramos optado por herramientas como WordPress o servicios como Ionos para obtener un dominio. Sin embargo, la página web se ve bien en diversos dispositivos y navegadores, y hasta ahora no hemos encontrado problemas de compatibilidad.

d) Aspectos de seguridad

registro.php: fichero comentado ya anteriormente, no se nos puede olvidar comentar el empleo de “password_hash” para el encriptado:

```
//Cifrado de contraseña muy importante.
$password_hash = password_hash($pass, PASSWORD_DEFAULT);
```


Es importante destacar que se ha implementado un proceso de encriptación para almacenar las contraseñas en la base de datos. Este paso de cifrado de contraseña es esencial para proteger la seguridad de las contraseñas de los usuarios. Utilizando algoritmos de hashing seguros, como el utilizado por "password_hash" con el parámetro "PASSWORD_DEFAULT", se garantiza que las contraseñas almacenadas en la base de datos no sean visibles en texto plano. Esto dificulta su lectura y aumenta significativamente la seguridad de los datos.

login.php: código también comentado anteriormente, pero no en el apartado de seguridad. Estas líneas de código definen el tiempo máximo de vida de la sesión y la caducidad de la cookie de sesión, asegurando que la sesión del usuario se cierre automáticamente después de un período de inactividad y que la cookie de sesión expire después de 30 minutos. Esto ayuda a mejorar la seguridad y la eficiencia del manejo de sesiones en la aplicación.

```
// Configurar el tiempo de vida de sesión
ini_set('session.gc_maxlifetime', 1800); // 1800 segundos = 30 minutos
// Configurar la caducidad de la sesión
session_set_cookie_params(1800); // 1800 segundos = 30 minutos
```

Muy importante configurar siempre nuestro fichero **php.ini** de manera adecuada, este se ubica en **/opt/lamp/etc**:

```
session.cookie_lifetime=86400
; http://php.net/session.cookie-lifetime
session.cookie_lifetime=86400
```

Seguridad ante SQL Injection

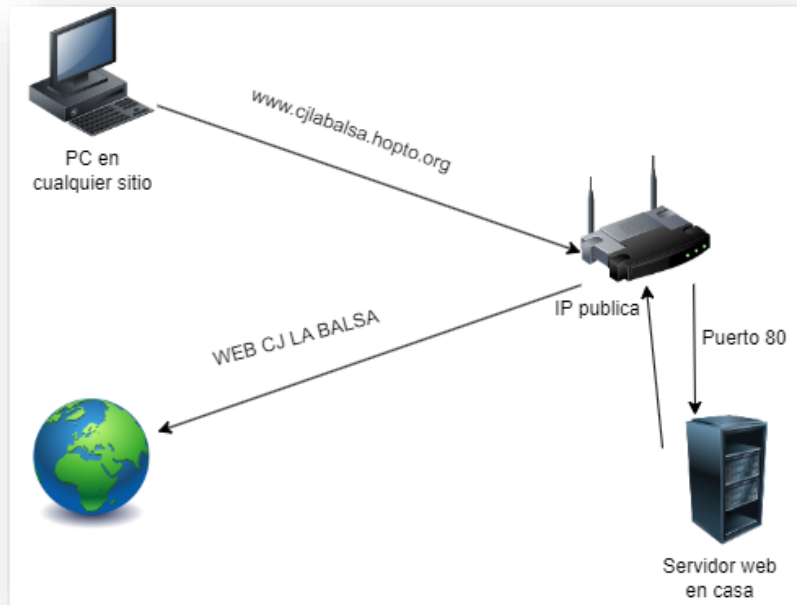
Además de la función **test_input()** que ya hemos visto durante nuestros estudios de ASIR, nos encontramos con una función propia de PHP que utilizamos en conjunto con la base de datos BBDD:

```
//Función IMPORTANTE ante el sql injection
$usuario = mysqli_real_escape_string($conn, $usuario);
```

Esta función, **mysqli_real_escape_string**, es utilizada para escapar los caracteres especiales en una cadena antes de ser insertada en una consulta SQL. Su propósito principal es prevenir la inyección SQL, que es un tipo de ataque en el cual un usuario malintencionado intenta manipular o comprometer la base de datos a través de entradas no validadas.

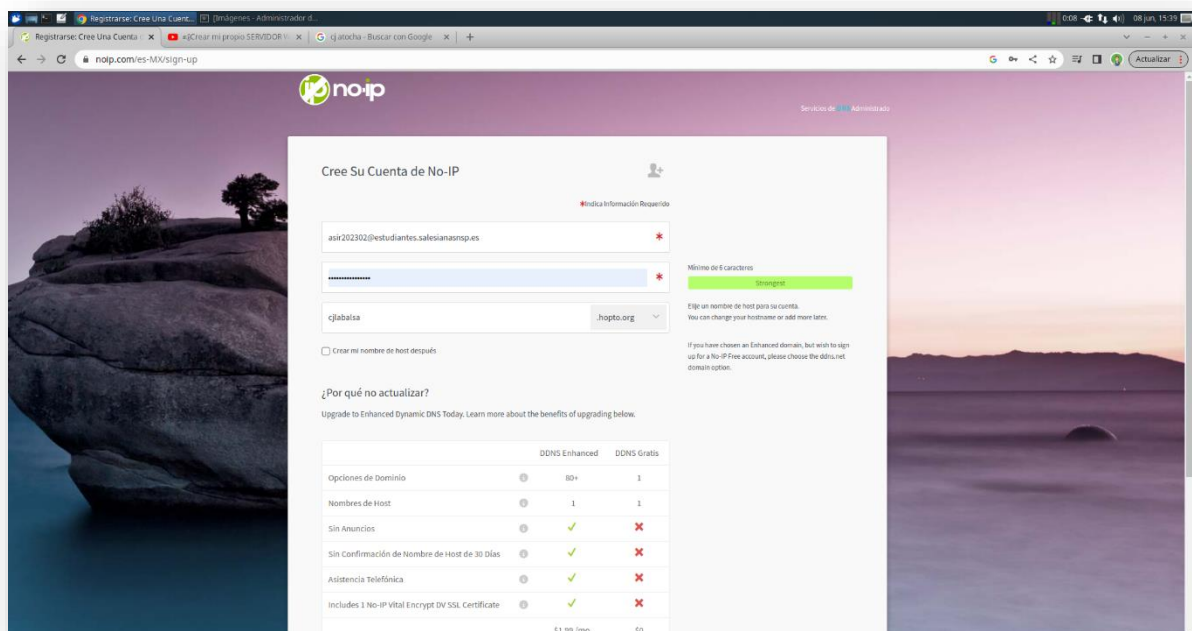
Al utilizar **mysqli_real_escape_string**, estamos asegurando que el valor de la variable **\$usuario** esté formateado adecuadamente para ser utilizado en una consulta SQL sin introducir errores o vulnerabilidades de seguridad.

Finalmente, después de completar la codificación de la página, llevamos a cabo el proceso de **acceso remoto** a la web alojada en nuestra máquina virtual siguiendo los siguientes pasos:



1. CREAR Y CONFIGURAR NUESTRO NOMBRE DDNS: <https://www.noip.com/>

- Dynamic DNS
- Create Hostname
- Cjlabalsa



Create Your No-IP Account

Email:
 Password:
 Hostname:
 Domain:

☐ Create my hostname later

¿Por qué no actualizar?

Upgrade to Enhanced Dynamic DNS Today. Learn more about the benefits of upgrading below.

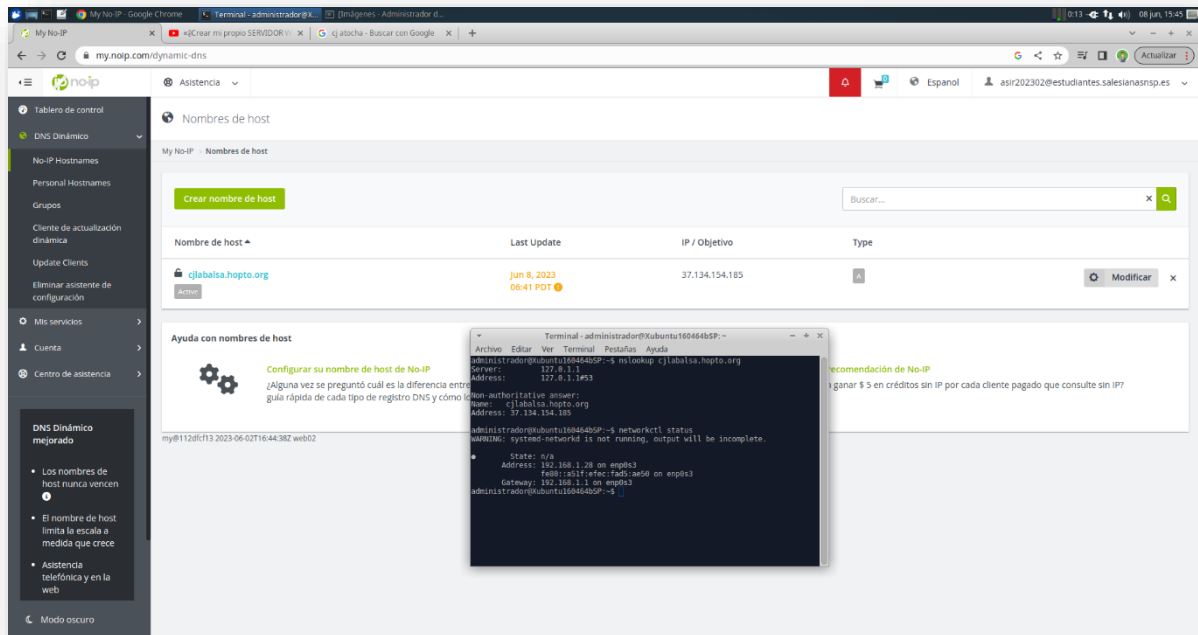
	DDNS Enhanced	DDNS Gratis
Opciones de Dominio	10+	1
Nombres de Host	1	1
Sin Anuncios	✓	✗
Sin Confirmación de Nombre de Host de 30 Días	✓	✗
Asistencia Telefónica	✓	✗
Incluye 1 No-IP Vital Encrypt DV SSL Certificate	✓	✗
	\$1.99 /mo	\$0

2. CONFIGURAR EL EQUIPO

- Máquina Virtual > Adaptador Puente

3. COMPROBAR LA RESOLUCIÓN DE NOMBRES DDNS

- Comprobar que tenemos Internet
- `networkctl status`
- `nslookup www.noip.com`



The screenshot shows the No-IP Dynamic DNS website interface. The left sidebar contains navigation links: Tablero de control, DNS Dinámico, No-IP Hostnames, Personal Hostnames, Grupos, Cliente de actualización dinámica, Update Clients, Eliminar asistente de configuración, Mis servicios, Cuenta, Centro de asistencia, and DNS Dinámico mejorado. The main content area is titled 'Nombres de host' and shows a table with one entry: 'cjabalsa.hopto.org' with a last update of 'Jun 8, 2023 06:41 PDT' and IP '37.134.154.185'. Below the table is a terminal window showing the output of the `nslookup` command, which returns the IP address '37.134.154.185' for 'cjabalsa.hopto.org'.

Nombre de host	Last Update	IP / Objetivo	Type
cjabalsa.hopto.org	Jun 8, 2023 06:41 PDT	37.134.154.185	Active

```
Terminal - administrador@Xubuntu1604645P:~$ nslookup cjabalsa.hopto.org
Server: 127.0.0.1
Address: 127.0.0.1
Non-authoritative answer:
Name: cjabalsa.hopto.org
Address: 37.134.154.185
```

4. CONFIGURAR EL ROUTER PARA QUE SE ACTUALICE

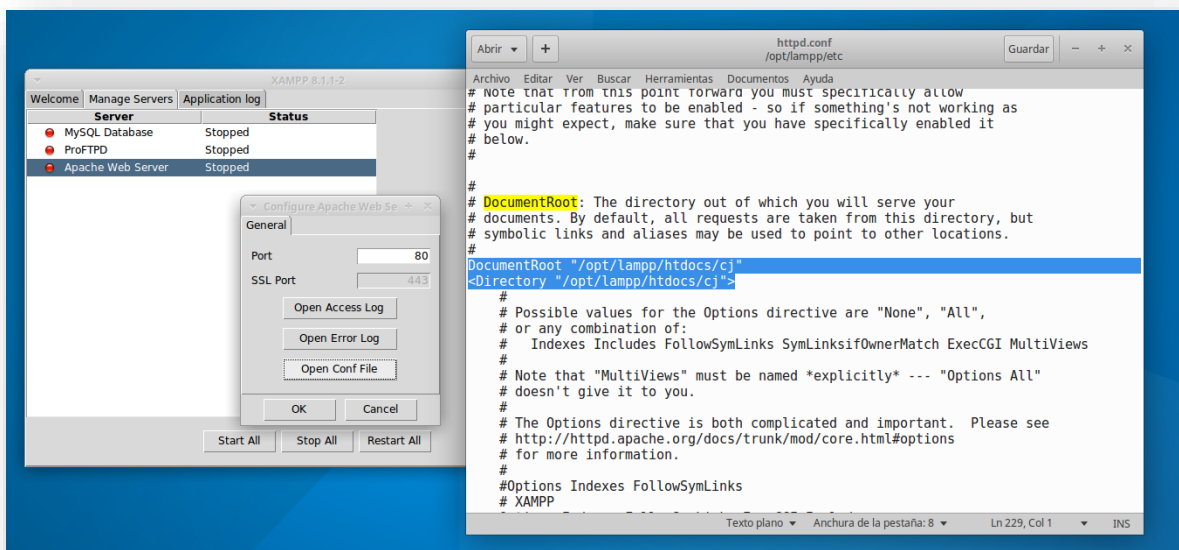
- Configuración del servidor de nombres dinámicos (DynDNS)



The screenshot shows the Livebox Fibra web interface. The top navigation bar includes links for 'mi red local', 'Wi-Fi', 'mis archivos', 'mi teléfono', 'información y diagnóstico', and 'configuración'. The left sidebar lists various configuration options, with 'configuración de la red' highlighted. The main content area is titled 'red' and contains a sub-menu with options like DHCP, NAT/PAT, DNS, UPnP, DynDNS, DMZ, NTP, and ONT. The DynDNS section is active, displaying a configuration form with fields for 'servicio' (set to dynDNS), 'nombre de host completo' (cjbahia.hopto.org), 'nombre de usuario email' (jesi202302@estudiantes.), and 'contraseña'. A table below the form shows the 'última actualización' (last update) as '2023/02/02 10:00:00'. The interface is in Spanish.

5. CONFIGURAR LA PÁGINA POR DEFECTO DE APACHE

- httpd.conf --> document root --> /opt/lampp/htdocs/cj/

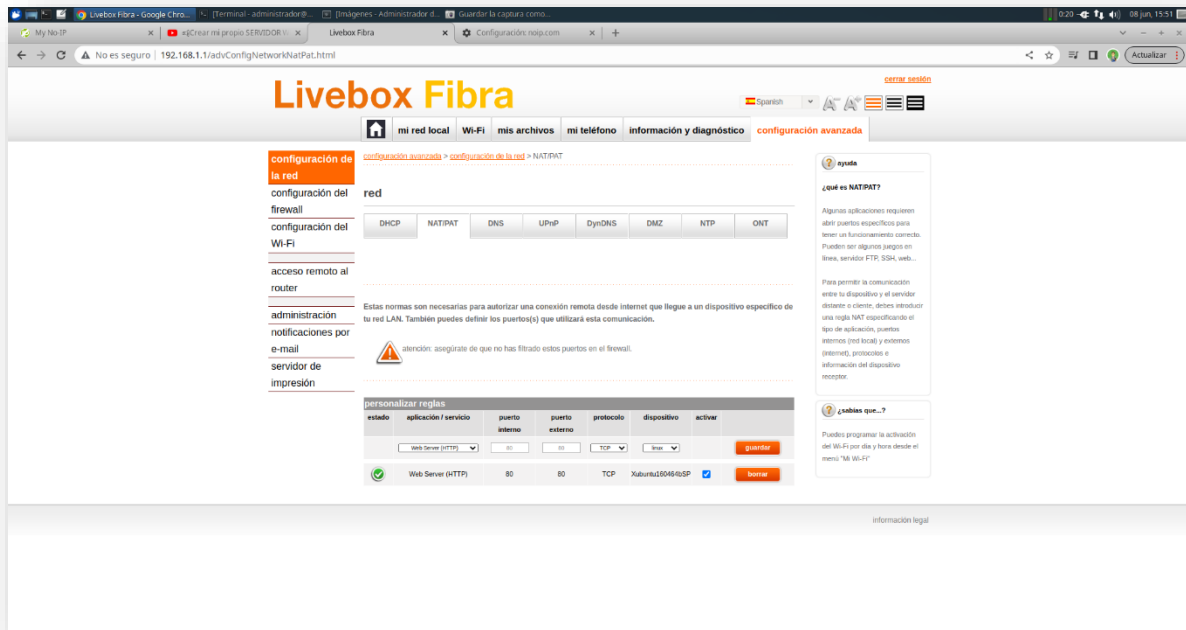


The screenshot shows the XAMPP 8.1.1-2 control panel on the left and the httpd.conf file editor on the right. The control panel shows the status of various services: MySQL Database (Stopped), ProFTPD (Stopped), and Apache Web Server (Stopped). The httpd.conf file is open, showing the 'DocumentRoot' and 'Directory' directives. The 'DocumentRoot' is set to '/opt/lampp/htdocs/cj/' and the 'Directory' is set to '/opt/lampp/htdocs/cj/'. The file editor shows the following configuration:

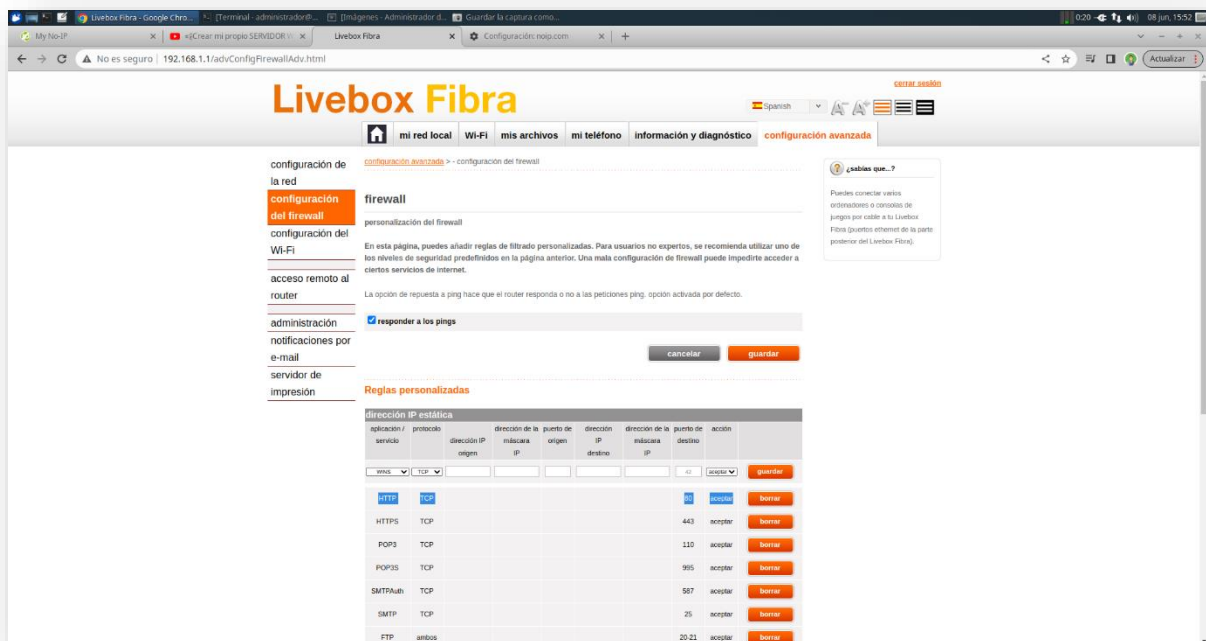
```
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/opt/lampp/htdocs/cj/"
<Directory "/opt/lampp/htdocs/cj/">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/trunk/mod/core.html#options
# for more information.
#
#Options Indexes FollowSymLinks
# XAMPP
```

6. CONFIGURAR EL ROUTER

- Configurar la tabla NAT del Router



- Configurar el Firewall



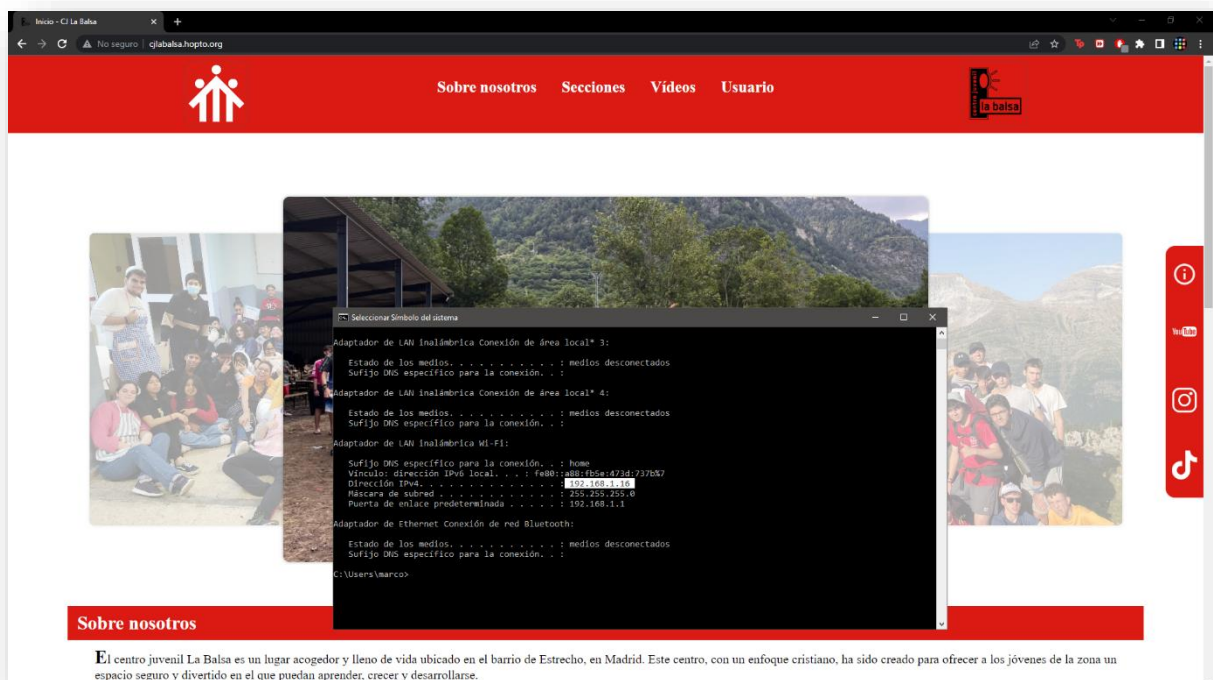
- Abrir puerto 80

```
GNU nano 2.5.3 Archivo: /etc/ufw/applications.d/http
[HTTP]
title=Web Server (HTTP)
description=Enable HTTP traffic
ports=80/tcp
```

```
administrador@Xubuntu160464bSP:~$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
administrador@Xubuntu160464bSP:~$ sudo ufw allow http
Regla añadida
Regla añadida (v6)
administrador@Xubuntu160464bSP:~$ sudo ufw status
Estado: activo

Hasta          Acción      Desde
-----
80             ALLOW      Anywhere
80 (v6)        ALLOW      Anywhere (v6)
```

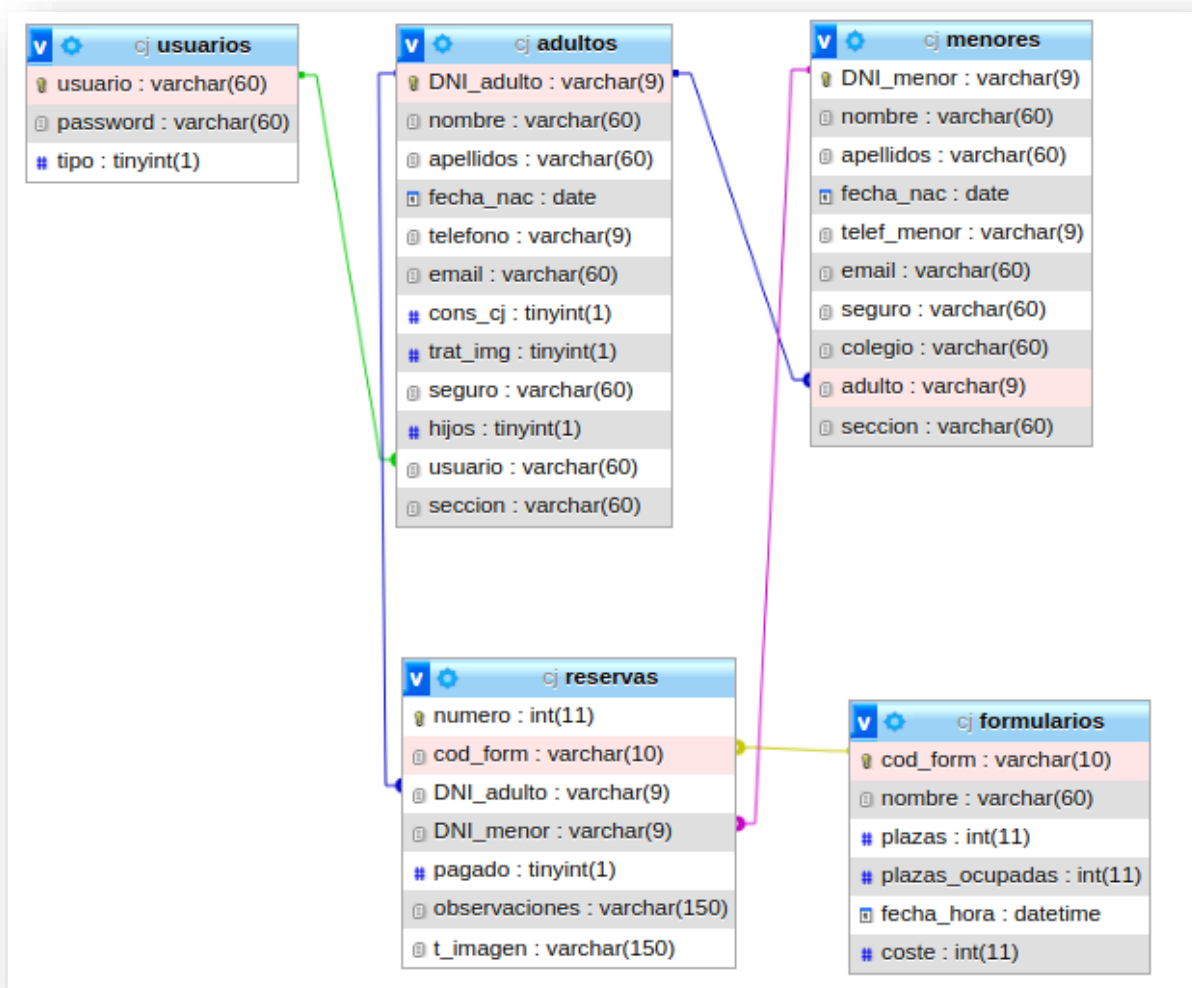
7. COMPROBAR EL FUNCIONAMIENTO DEL SERVICIO



Durante las pruebas realizadas, hemos notado que la página principal debe ubicarse siempre en la carpeta raíz para facilitar las rutas y la navegación entre carpetas. Inicialmente, encontramos dificultades al intentar acceder a "login.php" debido a que "index.php" se encontraba en la carpeta "main" (consultar el esquema de carpetas). Por lo tanto, en este momento hemos tomado la decisión de cambiar "index.php" a la carpeta "/cj/".



e) Diseño del modelo de datos



Nuestra base de datos ha sido diseñada cuidadosamente y se encuentra bien estructurada, lo que garantiza un óptimo rendimiento y eficiencia en el almacenamiento y acceso a los datos. Hemos seguido las mejores prácticas y estándares de diseño, asegurándonos de que las tablas estén correctamente relacionadas y que los campos estén debidamente definidos y normalizados.

Además, hemos implementado un esquema de seguridad sólido para proteger la integridad y confidencialidad de los datos almacenados. Se han aplicado técnicas de encriptación (**password**) y se han establecido adecuados niveles de permisos y autenticación para garantizar que solo los usuarios autorizados tengan acceso a la información sensible (**usuario administrador con su contraseña correspondiente**).

Además de su buen diseño, hemos realizado pruebas exhaustivas para garantizar la integridad y consistencia de la base de datos. Y gracias a ello es cuando nos damos cuenta de que la idea inicial que teníamos de la BBDD era errónea, comentado en las siguientes líneas:

musical.php:

En esta parte del proyecto, nos encontramos con un desafío relacionado con el diseño de la base de datos (BBDD). Originalmente, teníamos la idea de tener un campo llamado "DNI_participante" en la tabla de reservas que estaría vinculado a las tablas "DNI_menor" y "DNI_adulto" a través de dos relaciones foráneas. Sin embargo, al trabajar con phpMyAdmin, descubrimos que no era posible saltarse la restricción que nos obligaba a elegir un único valor para el campo "DNI_participante" que perteneciera a ambas tablas.

Compra de entradas para el musical

DNI Menor	Nombre
51511759Q	Ruben

Se ha producido un error al hacer una consulta del usuario en la base de datosError: INSERT INTO reservas (cod_form, DNI_participante, pagado, observaciones, t_imagen) VALUES ('MUS_1','51511759Q','1','','645b7839abd46_tux.jpg')
Cannot add or update a child row: a foreign key constraint fails ('cj`.`reservas`, CONSTRAINT `reservas_ibfk_1` FOREIGN KEY (`DNI_participante`) REFERENCES `adultos` (`DNI_adulto`) ON DELETE CASCADE ON UPDATE CASCADE)

DNI del menor:

Selecciona una representación:

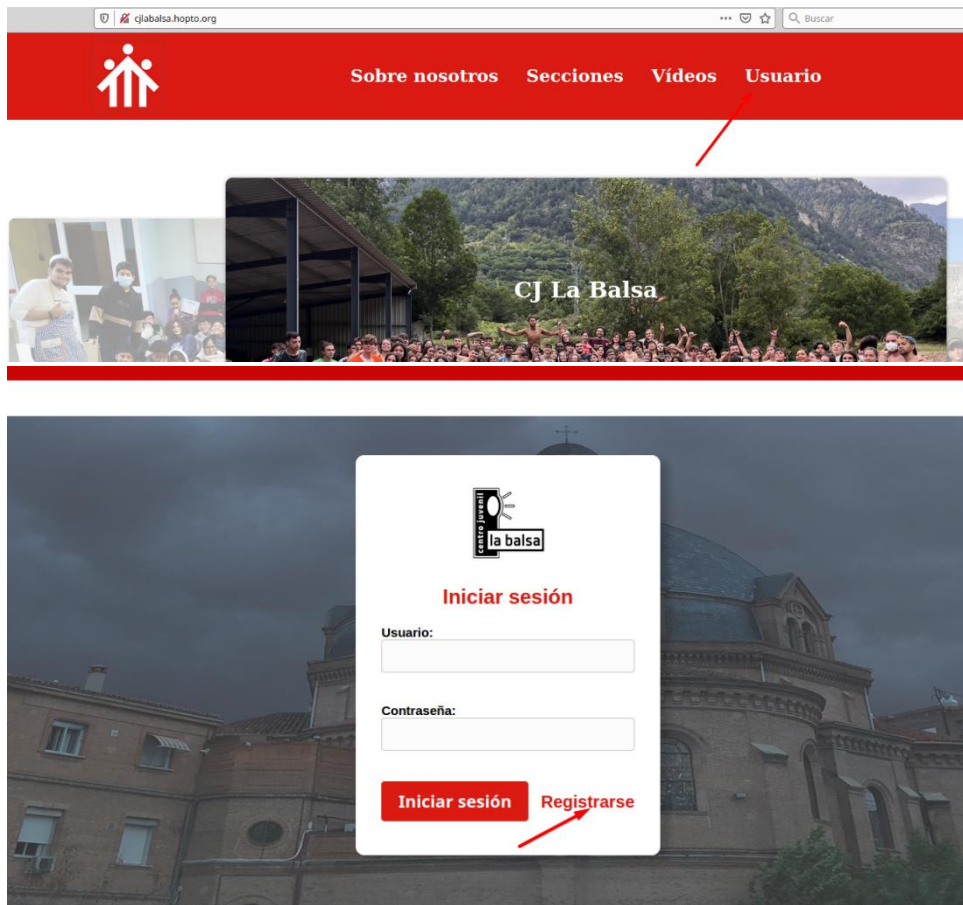
Imagen de transferencia: Ninguno archivo selec.


Observaciones:


Debido a esta limitación, nos dimos cuenta de que el enfoque inicial de nuestra BBDD no era viable. Tuvimos que replantear nuestra estructura y buscar una solución alternativa para gestionar la relación entre los participantes menores de edad y los adultos en la tabla de reservas. Por ello acabamos creando un campo número (clave primaria de la tabla reservas), y dos campos DNI_participante (por defecto NULL) y DNI_Adulto.


f) Diseño de la interfaz de usuario

En primer lugar, creamos un usuario realizando los siguientes clics:










Registro de Usuario


Usuario:


Contraseña:

Repite Contraseña:

¿Tienes hijos?
☒ si ☐ no







Registro de Usuario

DNIadulto:

Nombre:

Apellidos:

Fecha nacimiento:

Telefono:

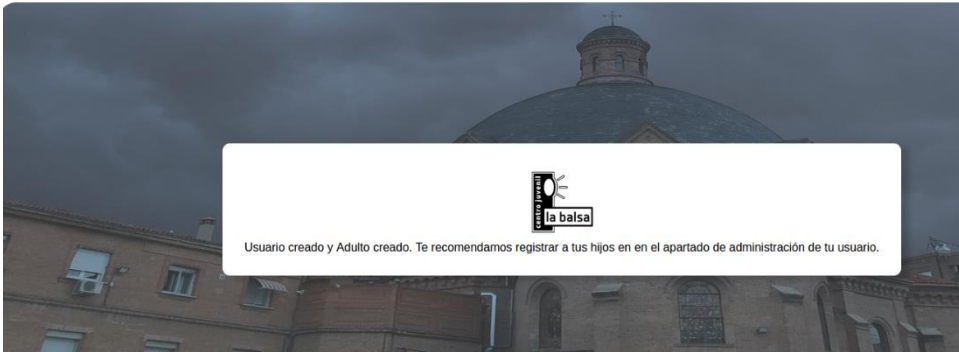
Email:


Consentimiento de participacion en el centro juvenil ☐

Consentimiento de uso y tratamiento de imágenes ☐

Seguro: ☐ Privado ☐ Publico ☐ Privado y Público

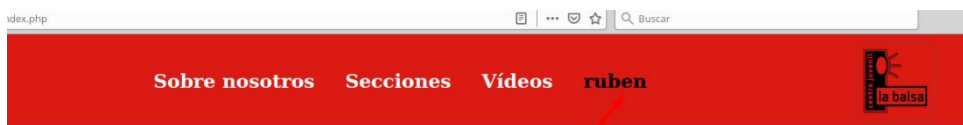
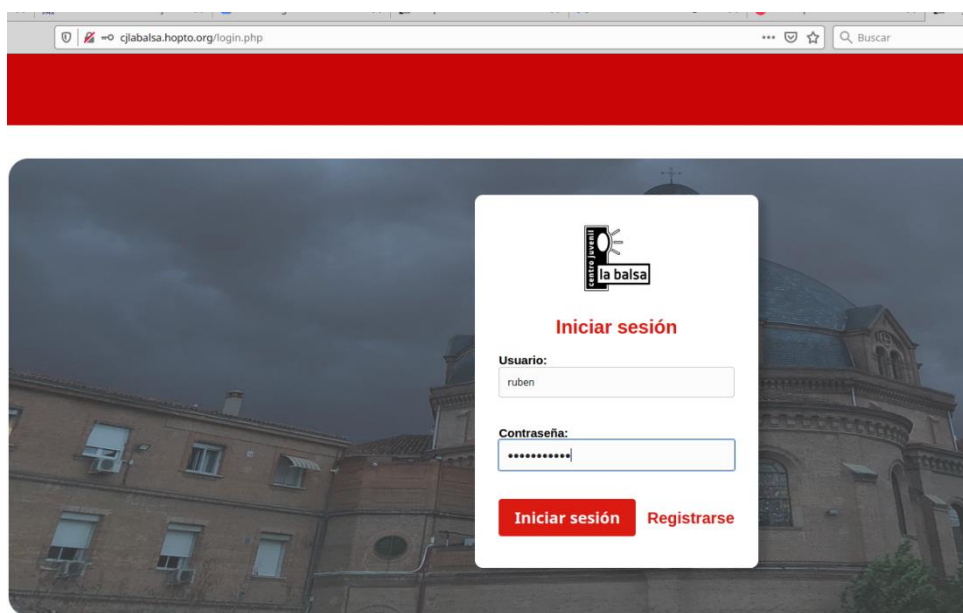
¿Cuantos hijos tienes?:



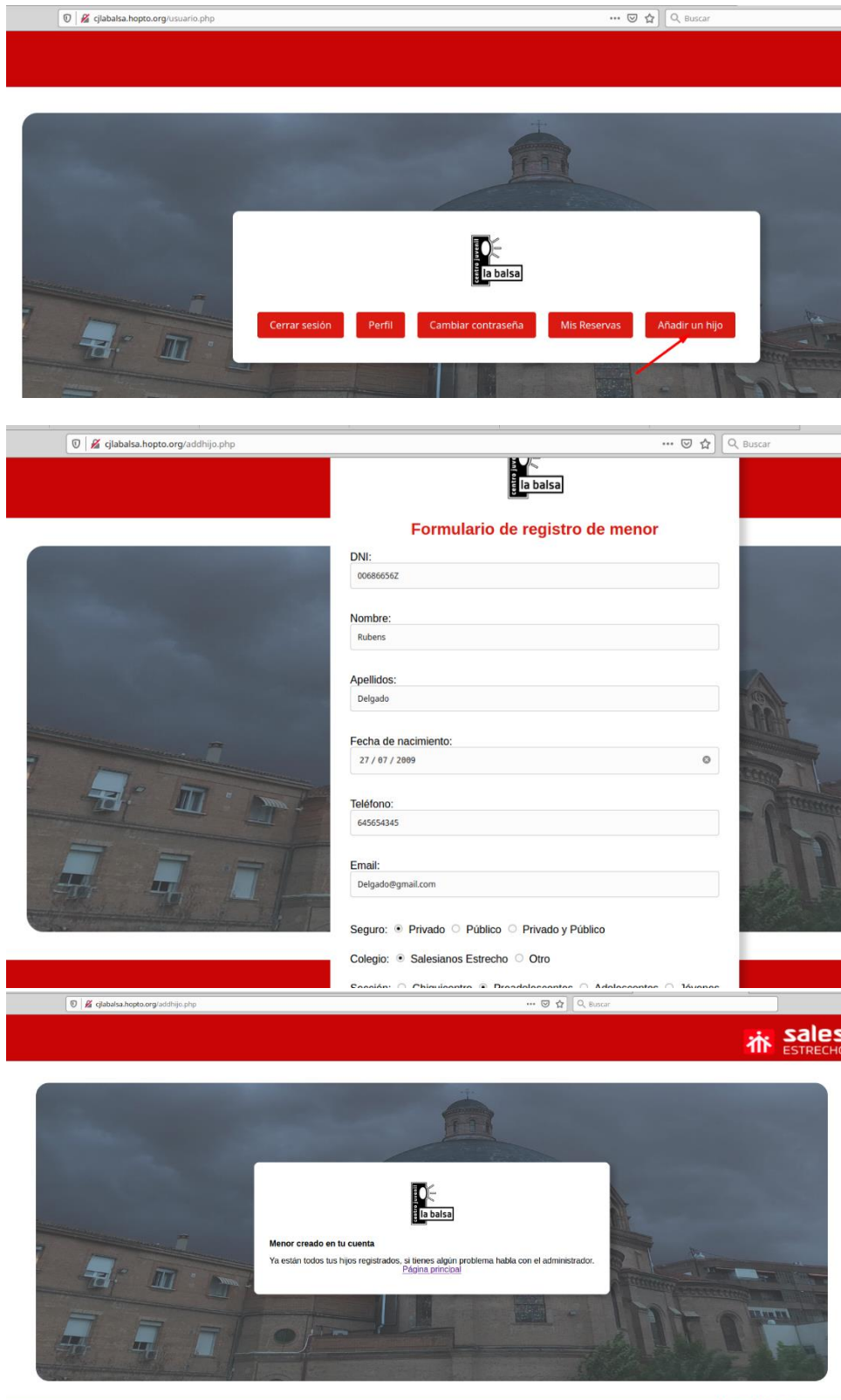


Usuario creado y Adulto creado. Te recomendamos registrar a tus hijos en en el apartado de administración de tu usuario.

Una vez ya realizado el registro iniciamos sesión:



Y ahora nos encontramos el área de administración del usuario. Vamos a añadir un hijo:



The image shows a sequence of three screenshots from a web browser, illustrating the process of adding a new user (child) to the system.

Top Screenshot: The browser address bar shows `cjlabalsa.hepto.org/usuario.php`. A white modal box is centered on the page, displaying the 'la balsa' logo and a navigation menu with the following buttons: 'Cerrar sesión', 'Perfil', 'Cambiar contraseña', 'Mis Reservas', and 'Añadir un hijo'. A red arrow points to the 'Añadir un hijo' button.

Middle Screenshot: The browser address bar shows `cjlabalsa.hepto.org/addhijo.php`. The page displays the 'Formulario de registro de menor' (Minor registration form). The form fields are as follows:


- DNI:** 00686656Z
- Nombre:** Rubens
- Apellidos:** Delgado
- Fecha de nacimiento:** 27 / 07 / 2009
- Teléfono:** 645654345
- Email:** Delgado@gmail.com
- Seguro:** ☒ Privado ☐ Público ☐ Privado y Público
- Colegio:** ☒ Salesianos Estrecho ☐ Otro
- Escuela:** ☐ Chiriquito ☐ Brandelencas ☐ Adalencas ☐ Bunas

Bottom Screenshot: The browser address bar shows `cjlabalsa.hepto.org/addhijo.php`. A white modal box is centered on the page, displaying the 'la balsa' logo and the following text:

Menor creado en tu cuenta
 Ya están todos tus hijos registrados, si tienes algún problema habla con el administrador.
[Página principal](#)

Realizamos un registro a una actividad:

No seguro | qlabalsa.hopto.org/formularios/cenaconfirmaciones.php



Registro Cena Confirmaciones 2023

DNI Menor	Nombre
00686656Z	Rubens

DNI del menor:


Imagen de transferencia: testfinal.PNG

Menú:

Consideraciones alimentarias:

Observaciones:

No seguro | qlabalsa.hopto.org/formularios/cenaconfirmaciones.php

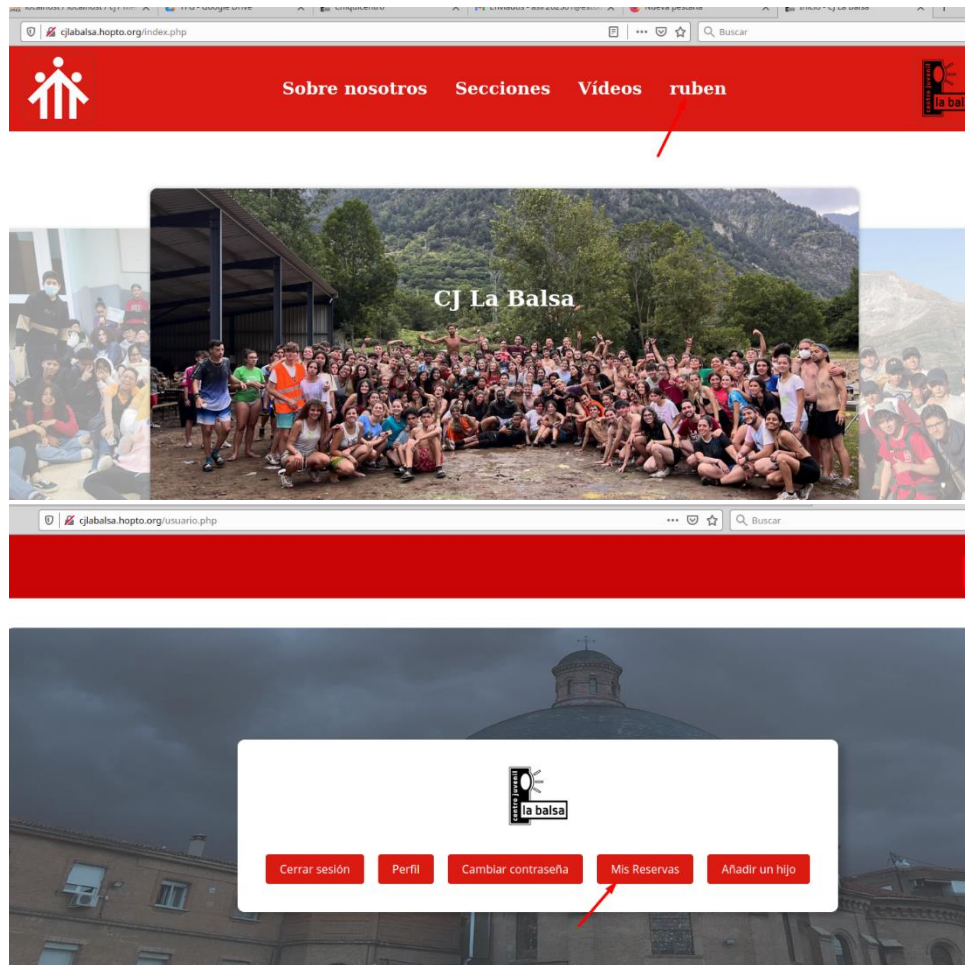


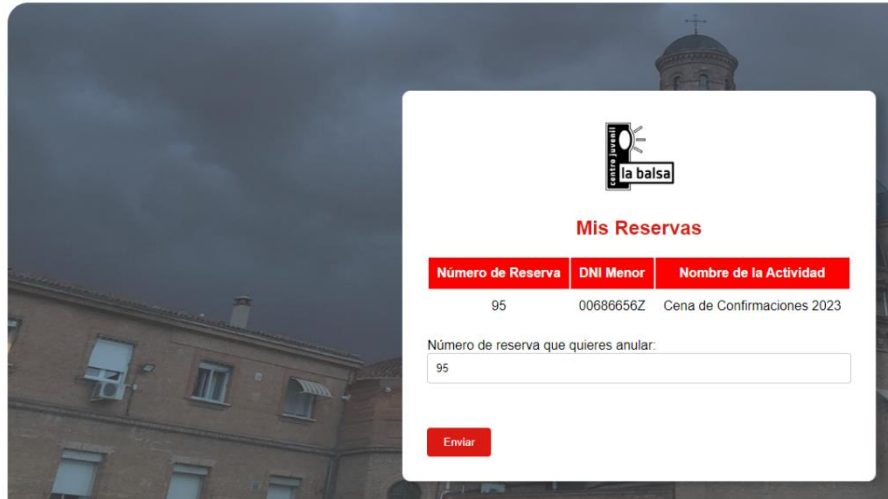
Registro Cena Confirmaciones 2023

DNI Menor	Nombre
00686656Z	Rubens

Registro realizado para la Cena.

Consultamos las reservas realizadas y anulamos:





Número de Reserva	DNI Menor	Nombre de la Actividad
95	00686656Z	Cena de Confirmaciones 2023

Número de reserva que quieres anular:

Enviar



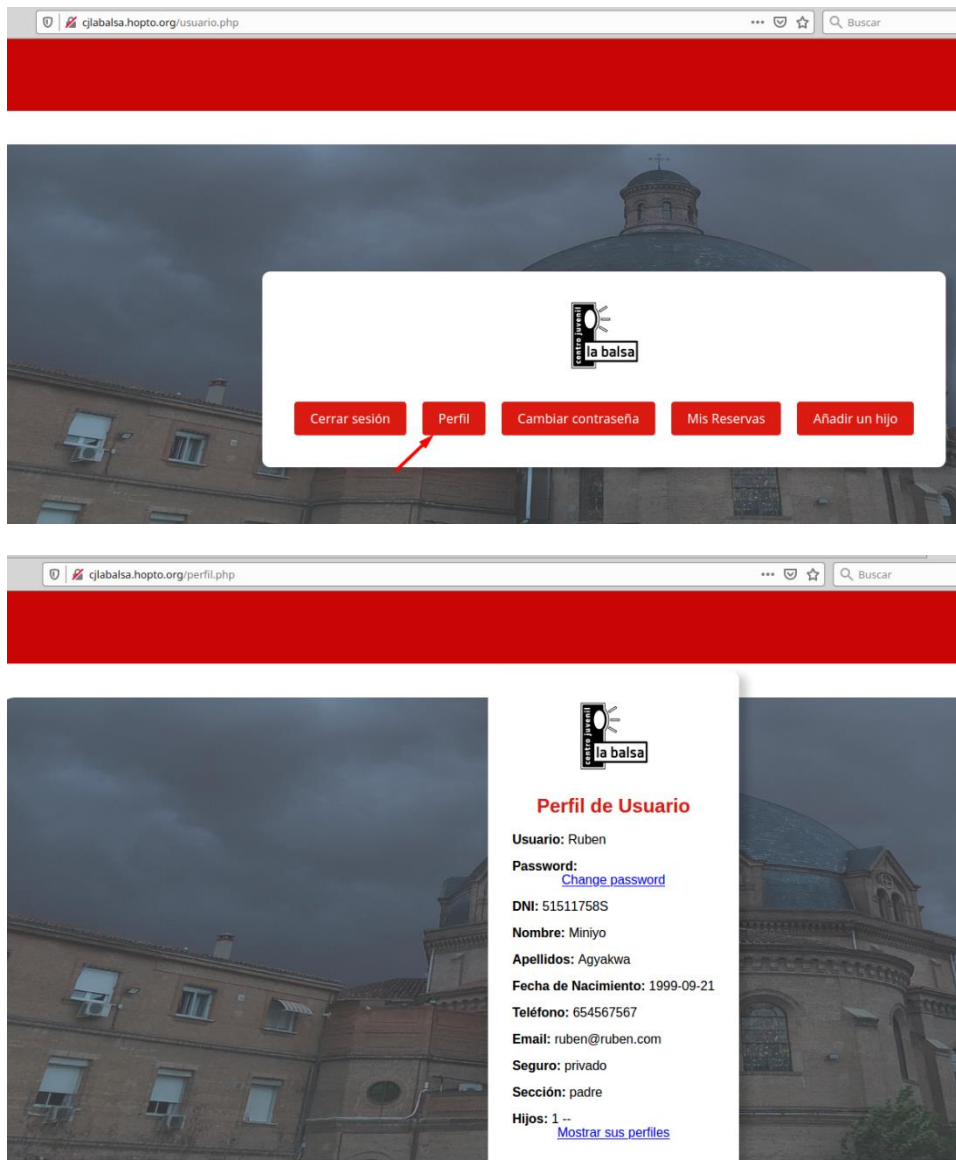
Número de Reserva	DNI Menor	Nombre de la Actividad
95	00686656Z	Cena de Confirmaciones 2023

Ha sido cancelada la reserva

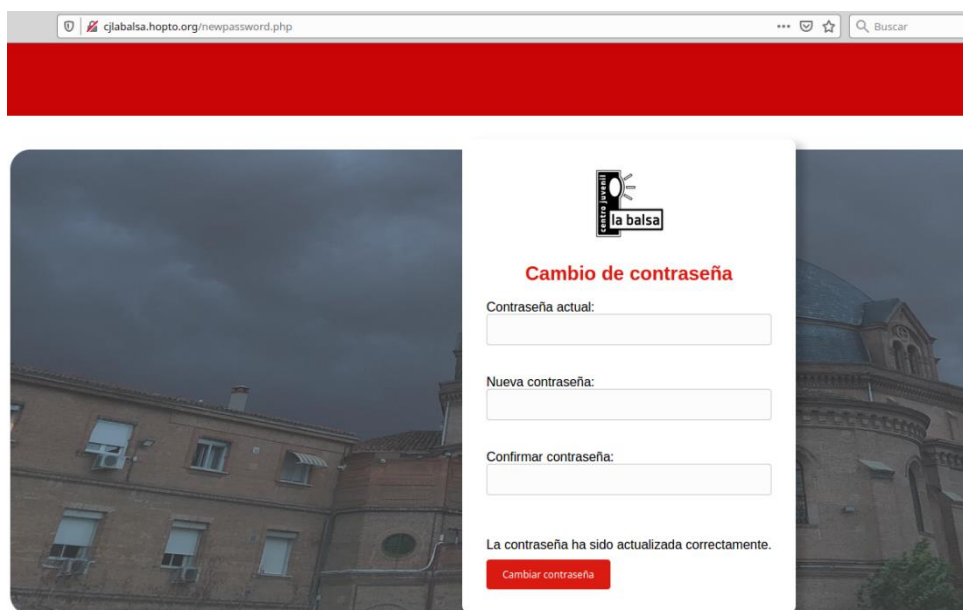
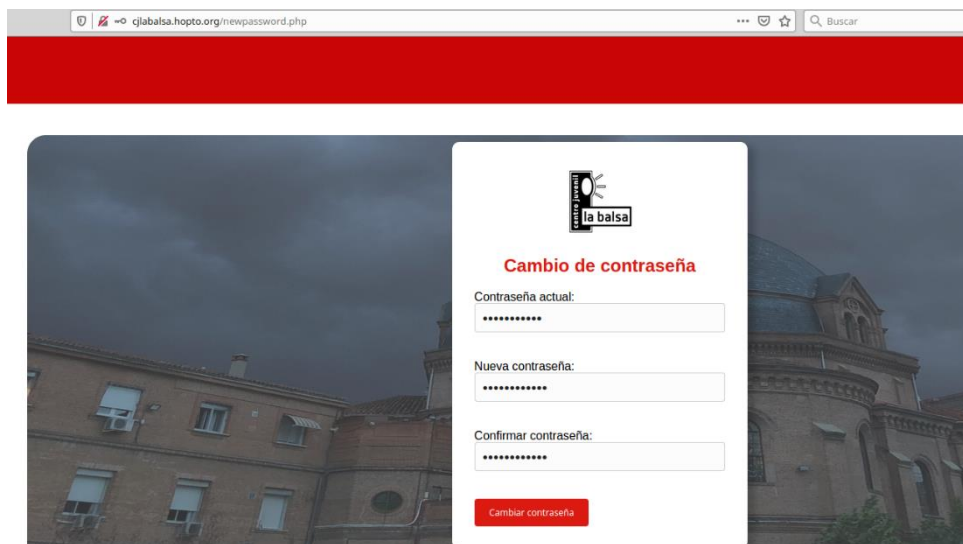
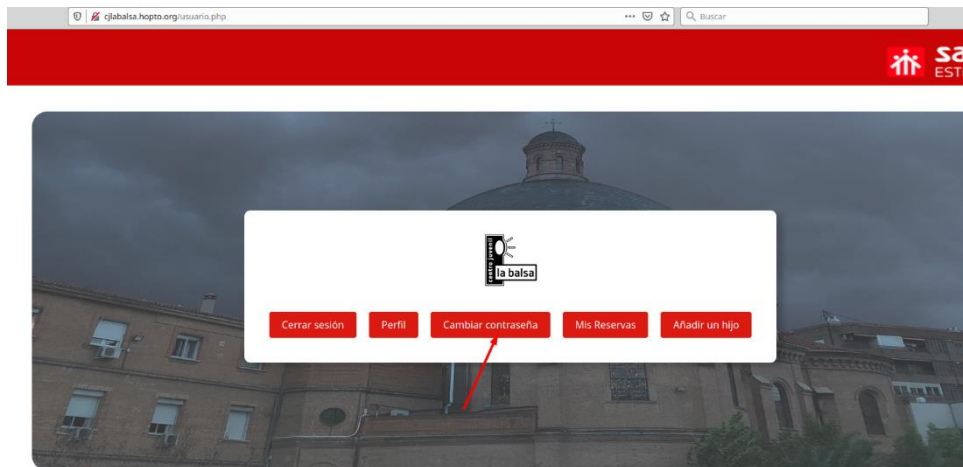
Número de reserva que quieres anular:

Enviar

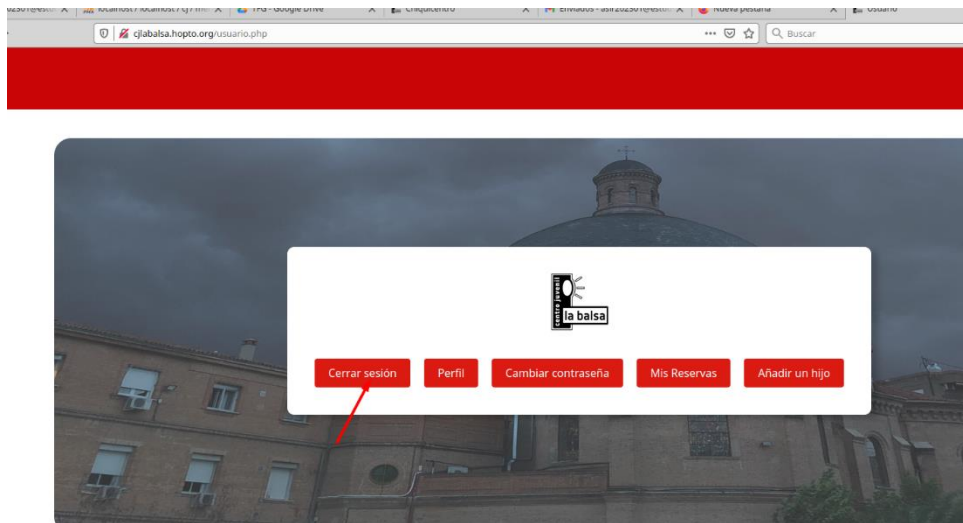
Consultamos nuestro perfil:



Realizamos un cambio de contraseña:



Y cerramos sesión:



IV. PLAN DE PRUEBAS

Hemos llevado a cabo diversas **pruebas internas** para evaluar el funcionamiento de la página web que hemos desarrollado. Durante este proceso, hemos puesto a prueba la codificación de HTML, CSS, PHP y JavaScript para asegurarnos de que todo esté en orden y funcionando correctamente. Algunas de las pruebas que hemos realizado incluyen:

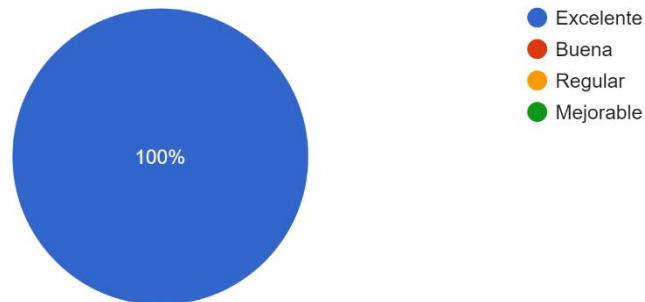
- Verificación de la estructura HTML: Hemos revisado minuciosamente el código HTML para garantizar que la estructura de la página esté bien definida y cumpla con los estándares.
- Estilo y diseño con CSS: Hemos comprobado la presentación visual de la página mediante la aplicación de estilos CSS. Nos hemos asegurado de que los elementos estén correctamente alineados, los colores y fuentes sean coherentes y el diseño sea atractivo.
- Interacción con JavaScript: Hemos realizado pruebas para verificar el correcto funcionamiento de cualquier interacción o animación implementada con JavaScript. Esto incluye la validación de formularios, la respuesta a eventos del usuario y la manipulación dinámica de elementos de la página.
- Funcionalidad de PHP: Hemos llevado a cabo pruebas exhaustivas de las funcionalidades implementadas en PHP, como el almacenamiento de datos en la base de datos, la gestión de usuarios y la generación de contenido dinámico. Nos hemos asegurado de que todas las operaciones se realicen de manera eficiente y segura.

Durante estas pruebas internas, hemos identificado y corregido errores o fallos en el código para mejorar la experiencia del usuario y garantizar el correcto funcionamiento de la página web. Además también hemos llevado a cabo pruebas de red para verificar el funcionamiento de la página web a través del acceso remoto (documentadas anteriormente).

También hemos realizado **pruebas externas**, enviando el link de la página web a conocidos y gente que pertenece al centro juvenil junto con una encuesta de satisfacción. El resultado de varias personas realizando peticiones a la web no ha dado ningún problema, y en general la satisfacción de la gente es buena. Adjuntamos gráficos:

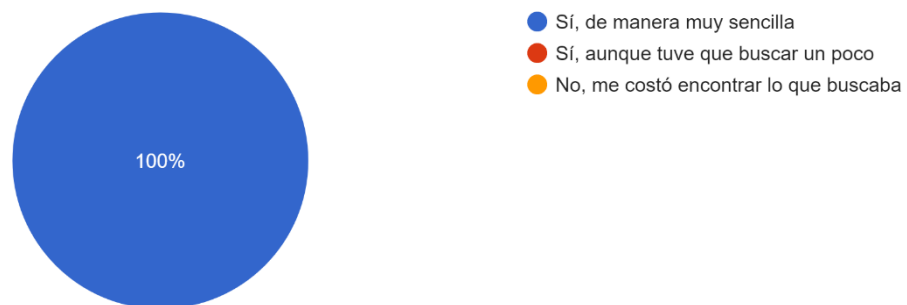
¿Qué te ha parecido la apariencia visual de nuestra página web?

5 respuestas



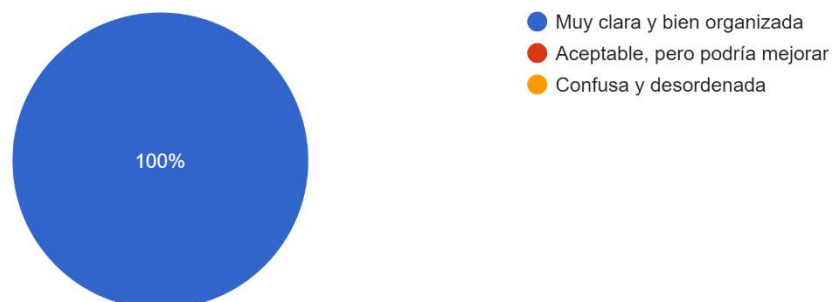
¿Encontraste fácilmente la información que buscabas?

5 respuestas



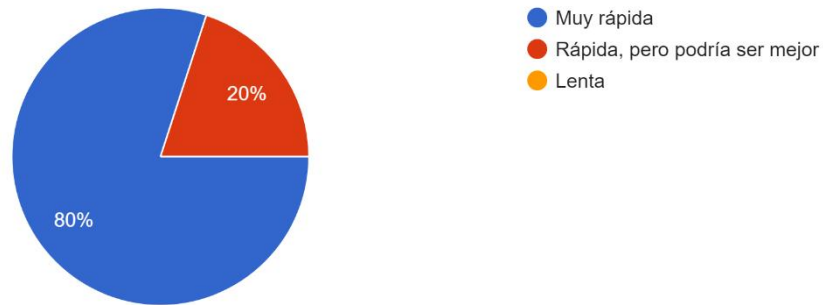
¿Qué opinas de la organización y estructura de la página web?

5 respuestas



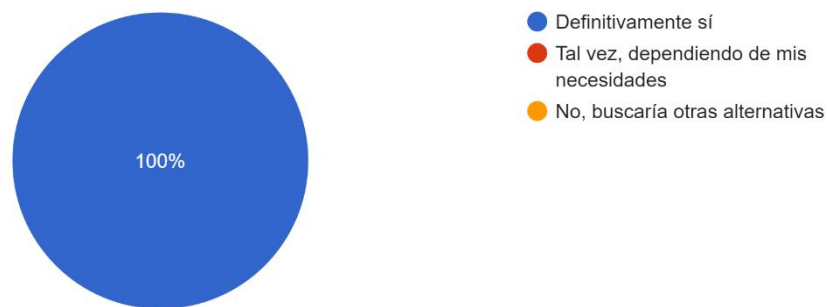
¿Qué opinas de la velocidad de carga de la página web?

5 respuestas



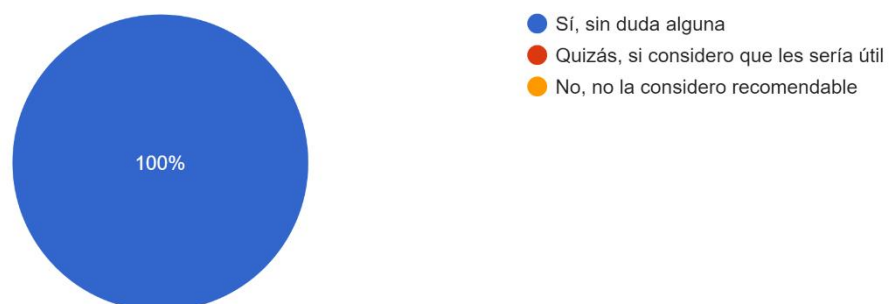
¿Utilizarías nuevamente nuestra página web en el futuro?

5 respuestas



¿Recomendarías nuestra página web a otras personas?

5 respuestas



¿Has tenido alguna experiencia positiva/negativa al completar formularios en nuestra página web?
Si es así, por favor descríbela brevemente.

5 respuestas

Es bastante intuitivo y al estar dentro de las actividades de cada nivel es muy fácil encontrarlos.

No se podía poner los apellidos acentuados

Ha sido súper rápido y bastante claro a la hora de cumplimentarlos

no.

Muy bonita

¿Tienes algún comentario adicional o sugerencia para mejorar nuestra página web?

4 respuestas

Como sugerencia en vídeos se podría poner la lista de youtube con los musicales

Nop

si.

En el apartado de Jóvenes y A2, sería recomendable poner ADOS ya que en la otra opción parece que hacer referencia al nivel concreto dejando de lado a A1

V. CONCLUSIONES

1. Objetivos cumplidos

Hemos logrado cumplir los objetivos establecidos al desarrollar una página web que satisface las principales necesidades de un Centro Juvenil salesiano. Nuestra página web proporciona información general sobre el centro, permite la realización de formularios para actividades, excursiones y campamentos, almacenando estos datos en una base de datos. Además, hemos implementado un sistema de gestión de usuarios confiable y redundante.

Ampliación de conocimientos: Nos hemos enfocado en el aprendizaje y la adquisición de nuevos conocimientos en las áreas de Aplicaciones Web y Bases de Datos. A través de nuestro proyecto con la práctica en un entorno real, hemos ampliado nuestra comprensión y dominio de estos campos.

Aplicación práctica: Hemos sido capaces de aplicar los conocimientos adquiridos de manera práctica en un entorno real. A través del desarrollo de nuestra idea, hemos puesto en práctica los conceptos y habilidades aprendidos, fortaleciendo nuestra capacidad para resolver problemas y enfrentar desafíos reales.

2. Conclusiones

Aunque nos hubiera gustado hacer mucho más y ofrecer una página de mayor calidad, reconocemos que dentro de nuestras posibilidades y limitaciones de tiempo y conocimiento, hemos dado lo mejor de nosotros. Nos damos cuenta de que este proyecto está más enfocado para estudiantes de DAW, pero hemos hecho todo lo que hemos podido para cumplir con los requisitos establecidos.

Durante el proceso, hemos ampliado nuestros conocimientos en las áreas de Aplicaciones Web y Bases de Datos. Hemos dedicado tiempo y esfuerzo en aprender y adquirir nuevas habilidades, y hemos logrado aplicar estos conocimientos en un entorno práctico y real. Esta experiencia nos ha permitido fortalecer nuestras habilidades de resolución de problemas y enfrentar desafíos reales.

Aunque nos hemos enfrentado a dificultades a lo largo del proyecto, hemos perseverado y utilizado nuestros recursos de la mejor manera posible. A pesar de las limitaciones, hemos logrado alcanzar nuestros objetivos y crear una página web funcional y útil para el Centro Juvenil. Estamos orgullosos de nuestro trabajo y de haber aprovechado esta oportunidad para aprender y crecer profesionalmente.

3. Futuras mejoras

Sería recomendable explorar la posibilidad de utilizar una plataforma adecuada, como WordPress, para desarrollar la página web en su totalidad. Con WordPress, sería posible codificar funcionalidades como la creación de tarjetas que presentan información sobre actividades y noticias de forma dinámica en lugar de estática. Esta elección proporcionaría una mayor flexibilidad y facilidad para actualizar y administrar el contenido de manera eficiente.

Contar con una adecuada financiación sería de gran ayuda para implementar un **TPV virtual** en la página que hemos creado, especialmente en la sección de registro de actividades, permitiendo a los usuarios realizar los pagos de manera segura y conveniente. La integración de un TPV virtual nos brindaría la posibilidad de aceptar pagos en línea, facilitando el proceso de inscripción y mejorando la experiencia del usuario al ofrecer opciones de pago más flexibles. Con la financiación adecuada, podríamos invertir en esta función para optimizar aún más nuestra página y brindar un servicio más completo a los usuarios.

Podríamos realizar una **implementación** más sólida y una mejor **unificación de HTML, CSS y PHP** en nuestro proyecto, mejorando así la integración de estos elementos. Además, podríamos optimizar la codificación en PHP para lograr una mayor eficiencia en nuestro código. Mediante una implementación más cuidadosa y una codificación PHP más eficiente, podríamos garantizar una estructura coherente y un rendimiento mejorado en nuestra página web. Esto nos permitiría ofrecer una experiencia más fluida y una mayor calidad en el funcionamiento general del sitio.

Una mejora futura importante que debemos considerar es fortalecer la seguridad de nuestra página web como realizando la aplicación de medidas de autenticación más robustas y la realización de pruebas de seguridad regulares para identificar posibles vulnerabilidades.

VI. BIBLIOGRAFÍA

[Include y Require PHP](#)

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

<https://www.w3schools.com/>

<https://stackoverflow.com/questions/tagged/php>

[🖥 Crear mi propio SERVIDOR WEB REMOTO desde CASA 🏠 Ubuntu + APACHE](#)

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

<https://codepen.io/yuhomyan/pen/WNwGywp>