

Diseño de aplicativo web para la fundación trapitos con amor, un aporte a la comunidad

Karen Tatiana Hortua Hortua

Asesor

Edgar Antonio Diaz Bustos

Universidad Nacional Abierta y a Distancia UNAD

Escuela Ciencias Básicas, Tecnología e Ingeniería

Ingeniería de Sistemas

2025

Resumen

Se desarrolla una aplicación web que tendrá funciones de front-end como de back-end, con el objetivo de promover la Fundación Trapitos con Amor a nivel nacional e internacional. Esto facilitará la comunicación, el reconocimiento y la colaboración. El sistema contará con una base de datos MySQL para gestionar la información de los beneficiarios y las donaciones, y también habrá un plan de mantenimiento que incluirá actualizaciones de seguridad, optimización y la posibilidad de ampliar las funcionalidades.

Palabras Claves: Aplicativo web, Fundación Trapitos con Amor, Front-end, Back-end, Comunicación digital.

Abstract

A web application is developed that will have both front-end and back-end functions, with the objective of promoting the Trapitos con Amor Foundation nationally and internationally. This will facilitate communication, recognition and collaboration. The system will have a MySQL database to manage beneficiary information and donations, and there will also be a maintenance plan that will include security updates, optimization and the possibility of expanding functionalities.

Keywords: Web application, Trapitos con Amor Foundation, Front-end, Back-end, Digital communication.

Tabla de Contenido

Introducción	11
Justificación	12
Planificación y levantamiento.....	13
Planificación.....	13
Definición del Alcance.....	13
Identificación de Interesados.....	19
Cronograma	20
Presupuestos	23
Levantamiento de Requerimientos Funcionales	23
Reuniones con la fundación.....	23
Análisis de Fundaciones Colombianas	24
Levantamiento de Requerimientos No Funcionales.....	24
Rendimiento.....	24
Seguridad.....	24
Usabilidad.....	24
Estabilidad	24
Compatibilidad	24
Mantenibilidad.....	24
Documentación de Requerimientos	25

Documento de Especificación de Requerimientos del Software.....	25
Definición de contextos y toma de decisiones de diseño	25
Contextos.....	25
Toma de decisiones de diseño	26
Validaciones ATAM	26
Escenarios de Validación.....	26
Evaluación de Riesgos.....	27
Análisis de Trade-offs	27
Diseño de la arquitectura del software.....	28
Diagrama de flujo del aplicativo web	28
Descripción:.....	28
Diagrama de flujo de base de datos.....	30
Descripción:.....	30
Diagrama de flujo de software	31
Descripción:.....	31
Desarrollar el aplicativo web	32
Análisis.....	32
Programación (Cambiar en el cronograma)	33
Definición de drivers	33
Diseño	98

Diseño de aplicativo web	98
Página Principal.....	98
Ayuda.....	100
Testimonios	103
¿Qué hacemos?	104
Política y Privacidad, Términos y Condiciones.....	106
Base de datos	108
Software.....	111
Plan de Pruebas	114
Descripción:	114
Prueba de pregunta y respuesta	114
Prueba de formulario por formsubmit.....	114
Instalación del servidor node.js.	115
Prueba de subida de una parte del diseño y adjuntar la base de datos en línea.....	117
Pestaña Donar.....	118
Conclusiones	119
Descripción:	119
Conceptos.....	121
Aislamiento (Isolation):.....	121
AJAX:.....	121

Arquitectura Monolítica:	121
Atomicidad (Atomicity):	121
Base de Datos (MySQL):	121
Base de datos relacionales:	122
Consistencia (Consistency):	122
Datos no estructurados:	122
Donantes:	122
Durabilidad (Durability):	122
Gestión de Datos (PHP + MySQL en XAMPP):	123
Hash:	123
Node.js:	123
Pasarelas de pago:	123
POST:	123
Transacciones ACID:	124
Visitantes:	124
Voluntarios:	124
Referencias Bibliográficas	125

Lista de Tablas

Tabla 1 <i>Cronograma</i> -----	20
Tabla 2 <i>Equipos y Software</i> -----	23
Tabla 3 <i>Menú principal</i> -----	33
Tabla 4 <i>Portada #1</i> -----	36
Tabla 5 <i>Portada #2</i> -----	37
Tabla 6 <i>Información de actividades</i> -----	38
Tabla 7 <i>Detalles y Carrusel</i> -----	39
Tabla 8 <i>Comentarios</i> -----	46
Tabla 9 <i>Integración</i> -----	50
Tabla 10 <i>¿Quiénes Somos?</i> -----	52
Tabla 11 <i>Menú desplegable</i> -----	55
Tabla 12 <i>Testimonios</i> -----	58
Tabla 13 <i>¿Qué hacemos?</i> -----	60
Tabla 14 <i>Donar</i> -----	62
Tabla 15 <i>Menú desplegable</i> -----	67
Tabla 16 <i>Parte principal formularios</i> -----	69
Tabla 17 <i>Explicación del Formulario</i> -----	70
Tabla 18 <i>Políticas y Privacidad, Términos y Condiciones</i> -----	74
Tabla 19 <i>Encabezado</i> -----	76
Tabla 20 <i>Usuario y Contraseña</i> -----	78
Tabla 21 <i>Tablas de la información registrada</i> -----	80
Tabla 22 <i>Servidor</i> -----	83

Tabla 23 <i>Ingresar contraseña nueva y usuario</i> -----	87
Tabla 24 <i>Conexión de base de datos</i> -----	89
Tabla 25 <i>Login Software</i> -----	91
Tabla 26 <i>Interfaz de la tabla</i> -----	92
Tabla 27 <i>Uso de dispositivo móvil</i> -----	94
Tabla 28 <i>Página Principal diseño</i> -----	99
Tabla 29 <i>Quienes Somos diseño</i> -----	101
Tabla 30 <i>Preguntas Frecuentes diseño</i> -----	102
Tabla 31 <i>Testimonios diseño</i> -----	103
Tabla 32 <i>¿Qué hacemos? diseño</i> -----	105
Tabla 33 <i>Política y privacidad diseño</i> -----	107
Tabla 34 <i>Contraseña y usuario</i> -----	108
Tabla 35 <i>Diseño de tablas base de datos</i> -----	109
Tabla 36 <i>Contraseña y usuario Software</i> -----	111
Tabla 37 <i>Diseño Tabla Software</i> -----	112
Tabla 38 <i>Prueba de pregunta y respuesta</i> -----	114
Tabla 39 <i>Prueba Formulario</i> -----	115
Tabla 40 <i>Instalación del servidor</i> -----	115
Tabla 41 <i>Prueba diseño y adjuntar la base de datos</i> -----	117
Tabla 42 <i>Prueba pestaña Donar</i> -----	118

Lista de figuras

Figura 1 <i>Diagrama ER (Entidad-Relación)</i>	14
Figura 2 <i>Diagrama UML</i>	15
Figura 3 <i>Arquitectura monolítica</i>	15
Figura 4 <i>Tablas de la base de datos</i>	16
Figura 5 <i>Diagrama Funcionamiento del servidor</i>	29
Figura 6 <i>Funcionamiento de la Base de datos</i>	30
Figura 7 <i>Manejo y Uso del Software</i>	31

Introducción

En la actualidad, la tecnología se ha convertido en una gran aliada para las organizaciones que buscan crecer, darse a conocer y fortalecer su labor social. La Fundación Trapitos con Amor, dedicada a brindar ayuda a quienes más lo necesitan, quiere contar con una herramienta digital que le permita conectarse mejor con las personas y mostrar su trabajo. Este proyecto tiene como propósito crear un aplicativo web que ayude a la fundación a fortalecer su presencia en línea y mejorar sus procesos de organización. A través de una plataforma con funciones front-end y back-end, se busca facilitar el acceso a la información, mantener una comunicación más cercana con la comunidad y contar con un espacio moderno, accesible y funcional. Además, se plantea un plan de mantenimiento que permita mejorar continuamente el sistema, garantizando su seguridad y adaptabilidad en el tiempo.

Justificación

La Fundación Trapitos con Amor necesita una herramienta tecnológica que le permita avanzar en su misión social. Muchas veces, las organizaciones sin ánimo de lucro enfrentan dificultades para gestionar su información, comunicarse efectivamente con su entorno o tener una presencia sólida en internet. Por eso, surge la necesidad de crear un aplicativo web que le ayude a organizar sus procesos internos y a mostrar su labor de una forma más clara y accesible. Este proyecto permitirá conectarse con más personas, dar a conocer su trabajo y fortalecer su identidad digital. Al contar con una plataforma hecha a la medida, se podrán mejorar aspectos como la visibilidad, la experiencia de quienes visitan la página y el crecimiento futuro de la organización. También se contempla un plan de mantenimiento para asegurar que la herramienta funcione correctamente y se adapte a las nuevas necesidades que puedan surgir.

Planificación y levantamiento

Planificación

Definición del Alcance.

Objetivo del proyecto

Diseñar, Desarrollar e implementar un aplicativo web con funciones front-end y back-end con el propósito de establecer un factor de comunicación a nivel nacional e internacional, que permita a los usuarios promover el conocimiento sobre la fundación trapitos con amor, así como fomentar el reconocimiento y la colaboración. Dentro del back-end se implementará una base de datos MySQL para gestionar de manera óptima la información de los beneficiarios de la fundación, así como los montos de donación. Se entregará un plan de mantenimiento para el aplicativo web y la base de datos, lo cual incluirá actualizaciones de seguridad, Dentro del back-end se implementará una base de datos MySQL para gestionar de manera óptima la información de los beneficiarios de la fundación, así como los montos de donación. Se entregará un plan de mantenimiento para el aplicativo web y la base de datos, lo cual incluirá actualizaciones de seguridad, optimización de consultas y la posibilidad de agregar nuevas funcionalidades.

Objetivos Específicos

Realizar la planificación y el levantamiento de requerimientos funcionales y no funcionales del aplicativo web con respecto a las necesidades de la organización.

Diseñar la arquitectura del software teniendo en cuenta el levantamiento de requerimientos de la fase anterior identificando así la mejor solución para la infraestructura TI existente.

Desarrollar el aplicativo web de acuerdo con la arquitectura diseñada a través de tecnología front-end y back-end con el fin de optimizar los procesos de gestión de donantes de la organización.

Ejecutar un plan de pruebas a través de la modelo cascada con el fin de identificar errores y mejorar el rendimiento el aplicativo web.

Proponer un plan de mantenimiento del aplicativo web con el fin de corregir problemas, supervisión de seguridad, rendimiento y experiencia de usuario.

Características Principales y Funcionalidad:

Página de inicio:

Propósito: Espacio de comunicación, publicidad e información.

Funcionalidades: Esta pestaña permitirá una visión integral de las actividades y eventos, incluyendo un pequeño párrafo sobre que hace la fundación, abordando un espacio para la ubicación y publicidad mediante videos que facilitarán la comprensión de su labor. También, se verá un formulario que permitirá recibir comentarios de los usuarios, y se incluirán enlaces a redes sociales y whatsapp, que permitirán acceder a ellas con solo hacer clic en los íconos correspondientes. Habrá un pequeño espacio donde se mostrarán imágenes de los voluntarios extranjeros participando en diversas actividades de la fundación.

Las funcionalidades descriptas en el proyecto están alineadas con los objetivos, a continuación, se presenta los diagramas detallados:

Diagrama ER (Entidad-Relación) para la base de datos

Figura 1

Diagrama ER (Entidad-Relación)

Diagramas UML descripción técnica

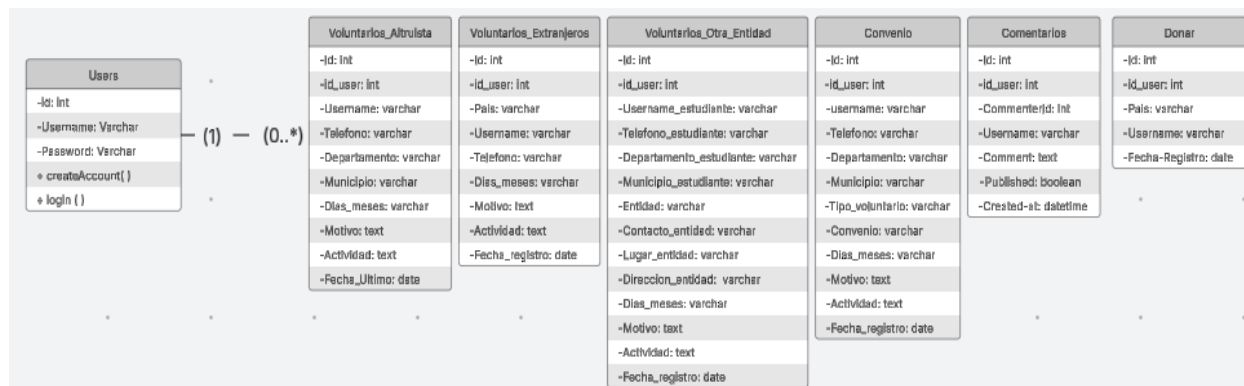


Figura 2

Diagrama UML

A continuación, se presenta un diagrama ER que describe las relaciones entre las tablas de la base de datos.

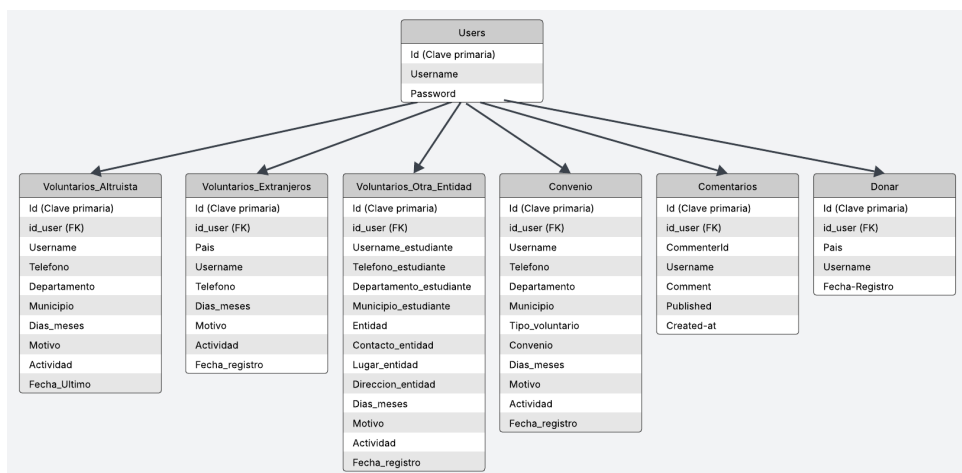


Figura 3

Arquitectura monolítica

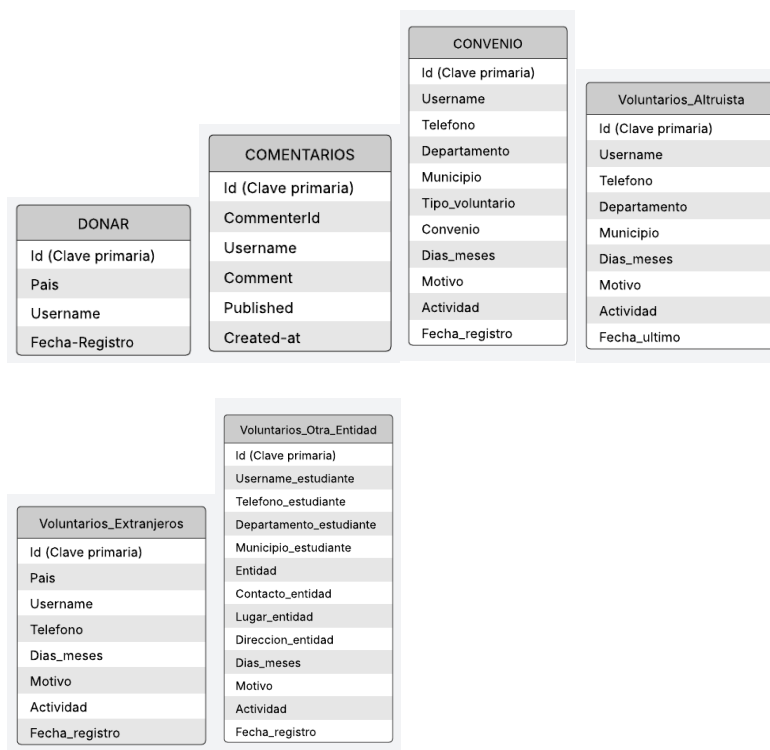
UI (HTML) Lógica (PHP / JS)



Base de datos

Figura 4

Tablas de la base de datos



Menú desplegable Ayuda

Botón ¿Quiénes somos?:

Propósito: Permitirá que los usuarios conozcan sobre la fundación.

Funcionalidades: Se presentará la vista de la historia, misión, visión, fundadores, colaboradores y pensamientos de la fundación. Cada una de estas secciones contará con un menú desplegable en el lado izquierdo y la información correspondiente en el lado derecho, ofreciendo así una visión más completa sobre la fundación.

Botón Preguntas Frecuentes:

Propósito: Permite conocer las preguntas y obtener información sobre alguna inquietud que se tenga.

Funcionalidades: Cada pregunta contará con un menú desplegable que, al ser seleccionado, permitirá desplegar y visualizar la respuesta correspondiente, facilitando así una experiencia interactiva y organizada para acceder a la información de manera clara y eficiente.

Botón Testimonios:

Propósito: Permitirá que los usuarios confíen en la fundación.

Funcionalidades: Habrá pantallazos y videos con mensajes de agradecimiento y cariño de las personas que han colaborado en la fundación, ya sea como ayudantes regulares, como aquellos que han finalizado su trabajo voluntario, o como beneficiarios de los servicios ofrecidos.

Botón ¿Qué hacemos?

Propósito: Permite observar ejemplos concretos de actividades realizadas, proyectos, casos de éxito y productos o servicios que ofrece la fundación.

Funcionalidad: Ofrecerá un detalle claro de cada proyecto, con un breve párrafo explicando lo que sucede en cada uno, acompañado de fotografías que mostrando las interacciones y las entregas realizadas ese día.

Voluntario

Propósito: Permite que los usuarios voluntarios llenen el formulario de inscripción.

Funcionalidad: En la sección se encontrará cuatro formularios diseñados para diferentes tipos de usuarios. Tres de estos formularios estarán dirigidos a personas residentes en Colombia, mientras que el cuarto estará destinado a usuarios extranjeros. Cada formulario contendrá una serie de preguntas breves que deben responder. Para completar el proceso, simplemente se deben llenar todos los campos requeridos y hacer clic en el botón de enviar. Este sistema está diseñado para recopilar la información necesaria de manera eficiente y organizada.

Donar

Propósito: Permitirá recibir los recursos para la fundación

Funcionalidad: Al ingresar, se presentará un formulario breve que, tras ser completado y enviado, mostrará un mensaje de agradecimiento. Luego, redirigirá al usuario al enlace del banco para realizar la donación. En lugar de mostrar los datos de la cuenta, el formulario solicitará la información del donante.

Políticas y privacidad, y Términos y Condiciones

Propósito: Una herramienta esencial tanto para la protección legal y privacidad del propietario del sitio web como para la información y transparencia con los usuarios.

Funcionalidad: Proporcionará información clara y accesible sobre cómo se recopilan, usan, almacenan y protegen los datos personales de los usuarios, asegurando que el sitio web cumpla con las leyes y regulaciones aplicables relacionadas con la privacidad y protección de datos, como el GDPR (Reglamento General de Protección de Datos) en Europa o la CCPA (Ley de Privacidad del Consumidor de California) en los Estados Unidos y Colombia. Proporcionará acceso al documento de términos y condiciones, asegurando que los usuarios comprendan y

acepten las reglas de uso del sitio, y mantendrá a los usuarios informados sobre cualquier cambio en estos términos.

Base de datos

BD. Aplicativo web:

Propósito: Permitirá que la fundación mejore la organización de los datos, seguridad y privacidad.

Funcionalidad: Se evidenciará un menú desplegable en el que, al hacer clic en alguna de las palabras que se encuentran allí, se dirigirá a la tabla correspondiente, permitiendo también ajustar y eliminar algún dato. Además, se encontrará al principio una pestaña para usuario y contraseña antes de entrar a la base de datos.

Software:

Propósito: Permite ingresar los datos de los beneficiarios de manera ordenada.

Funcionalidad: Contará con una pestaña para usuario y contraseña, de modo que solo las personas con permiso puedan acceder. Luego, en la siguiente pestaña, se mostrarán los diferentes datos ingresados en la tabla de beneficiarios, junto con botones para actualizar, eliminar, ingresar y buscar, con el fin de llevar un control y orden en los datos.

Identificación de Interesados

- **Director del proyecto:** Edgar Antonio Diaz Bustos
- **Desarrollador:** Karen Tatiana Hortua Hortua
- **Diseñadores:** Fundación trapitos con amor, Karen Tatiana Hortua Hortua e investigación.




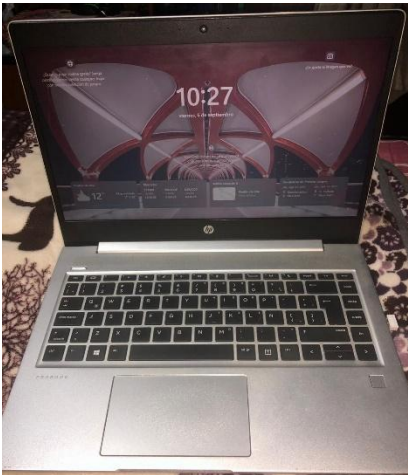
Ejecutar un plan de pruebas	
Comprender los objetivos.	
Determinar qué nuevas funcionalidades se desea testear	
Definir la estrategia	
Definir los criterios	
Establecer los entornos requeridos	
Planificación	

Presupuestos

- **Equipo humano:** 1,600 horas a 13,195 pesos por hora da un total \$21,112,000
- **Equipos y Software:** 1 computadora de 16gb con procesador Intel i5 tiene un costo de \$1,800,000.

Tabla 2

Equipos y Software

Configuración:	Portátil:														
<table border="1"> <tr> <td>Nombre del dispositivo</td> <td>DESKTOP-6U7K3F0</td> </tr> <tr> <td>Procesador</td> <td>Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz</td> </tr> <tr> <td>RAM instalada</td> <td>16,0 GB (15,9 GB usable)</td> </tr> <tr> <td>Identificador de dispositivo</td> <td>AC56D670-1F23-4777-A705-06E06ECB4CBD</td> </tr> <tr> <td>Id. del producto</td> <td>00330-52517-66885-AAOEM</td> </tr> <tr> <td>Tipo de sistema</td> <td>Sistema operativo de 64 bits, procesador basado en x64</td> </tr> <tr> <td>Lápiz y entrada táctil</td> <td>La entrada táctil o manuscrita no está disponible para esta pantalla</td> </tr> </table>	Nombre del dispositivo	DESKTOP-6U7K3F0	Procesador	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz	RAM instalada	16,0 GB (15,9 GB usable)	Identificador de dispositivo	AC56D670-1F23-4777-A705-06E06ECB4CBD	Id. del producto	00330-52517-66885-AAOEM	Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64	Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla	 
Nombre del dispositivo	DESKTOP-6U7K3F0														
Procesador	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz														
RAM instalada	16,0 GB (15,9 GB usable)														
Identificador de dispositivo	AC56D670-1F23-4777-A705-06E06ECB4CBD														
Id. del producto	00330-52517-66885-AAOEM														
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64														
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla														

Levantamiento de Requerimientos Funcionales

Reuniones con la fundación

Se realizarán llamadas telefónicas y mensajería para conocer las ideas que tienen, que impacto positivo les traerá esto a la fundación.

Análisis de Fundaciones Colombianas

Se revisa estos sitios web con el fin de generar buenas ideas y conocimientos para generar la creación del aplicativo.

Levantamiento de Requerimientos No Funcionales

Rendimiento

- El sitio debe cargar en menos de 4 segundos.
- Soportar al menos 500 usuarios concurrentes.

Seguridad

- Implementar HTTPS.
- Encriptar datos sensibles (por ejemplo, contraseñas y datos de pago).

Usabilidad

- Diseño adaptable para dispositivos móviles y computador.
- Interfaz intuitiva y fácil de navegar.

Estabilidad

Estructura modular que permita añadir nuevas funcionalidades en el futuro.

Compatibilidad

Soporte para los navegadores web más populares (Chrome, Firefox, Safari, Edge) y programas para descargar aplicaciones.

Mantenibilidad

- Código bien estructurado.

- Uso de buenas prácticas de desarrollo y patrones de diseño.

Documentación de Requerimientos

Documento de Especificación de Requerimientos del Software

- Wireframes en Balsamiq de las principales páginas del sitio web.
- Descripción detallada de cada requerimiento funcional y no funcional.
- Uso del servidor xampp y node.js para indicar que esté funcionando bien mi base de datos.
- Aplicación de neatbeans para la facilidad de uso de la base de datos.
- Aplicación sublime text para programación y diseño del aplicativo.
- Aplicación node.js para hacer la conexión con la base de datos y la aplicación web.

Definición de contextos y toma de decisiones de diseño

Contextos

Externo

Usuarios: Voluntarios, donantes y visitantes.

Sistemas datos: Base de Datos (MySQL), Sistema de Gestión de Datos (PHP + MySQL en XAMPP), Pasarelas de pago.

Dispositivos: Navegadores web, dispositivos móviles.

Interno

Front-end: Interfaz de usuario (HTML, CSS, JavaScript).

Back-end: Servidor web Apache XAMPP, programación PHP, JAVA, Base de datos MySQL y node.js.

Toma de decisiones de diseño

Arquitectura Monolítica (PHP y MySQL en XAMPP)

Pros: Simplicidad, menor costo inicial de desarrollo, fácil de implementar y desplegar con XAMPP.

Contras: Menor escalabilidad, dificultad para mantener y actualizar, riesgos de fallo único.

Decisión: Arquitectura Monolítica. Dada la simplicidad y facilidad de implementación en XAMPP, una arquitectura monolítica es adecuada para un proyecto inicial con recursos limitados.

Base de datos MySQL (utilizando XAMPP)

Pros: Transacciones ACID, soporte robusto para relaciones complejas, ampliamente soportado en PHP.

Contras: Menor flexibilidad para datos no estructurados.

Decisión: MySQL. La elección de MySQL se alinea bien con el uso de XAMPP y los requisitos de integridad y relaciones de datos.

Validaciones ATAM

Escenarios de Validación

Escalabilidad: Pruebas de carga con herramientas como JMeter para simular 10,000 usuarios simultáneos y verificar el rendimiento.

Seguridad: Realizar pruebas de penetración para evaluar las medidas de seguridad (implementación de SSL/TLS y autenticación de dos factores).

Disponibilidad: Realizar pruebas de resistencia para asegurar que el sistema cumple con el SLA de disponibilidad del 99.9%.

Evaluación de Riesgos

Escalabilidad de la Arquitectura Monolítica: Riesgo de que la aplicación no escale adecuadamente con un gran número de usuarios.

Seguridad: Riesgo de vulnerabilidades en la autenticación y el manejo de datos sensibles.

Mitigación: Implementar HTTPS, autenticación de dos factores y realizar auditorías de seguridad regulares.

Análisis de Trade-offs

Simplicidad vs. Escalabilidad: Elegir una arquitectura monolítica simplifica el desarrollo inicial, pero puede requerir refactorización a una arquitectura más distribuida a medida que el sistema crezca.

Robustez de MySQL vs. Flexibilidad de NoSQL: MySQL proporciona robustez y transacciones ACID, lo cual es crucial para la integridad de los datos en un sistema de comercio electrónico.

Diseño de la arquitectura del software

Diagrama de flujo del aplicativo web

Descripción:

En este diagrama podemos observar el funcionamiento del servidor en los formularios y la visualización del aplicativo web.

Figura 5

Diagrama Funcionamiento del servidor

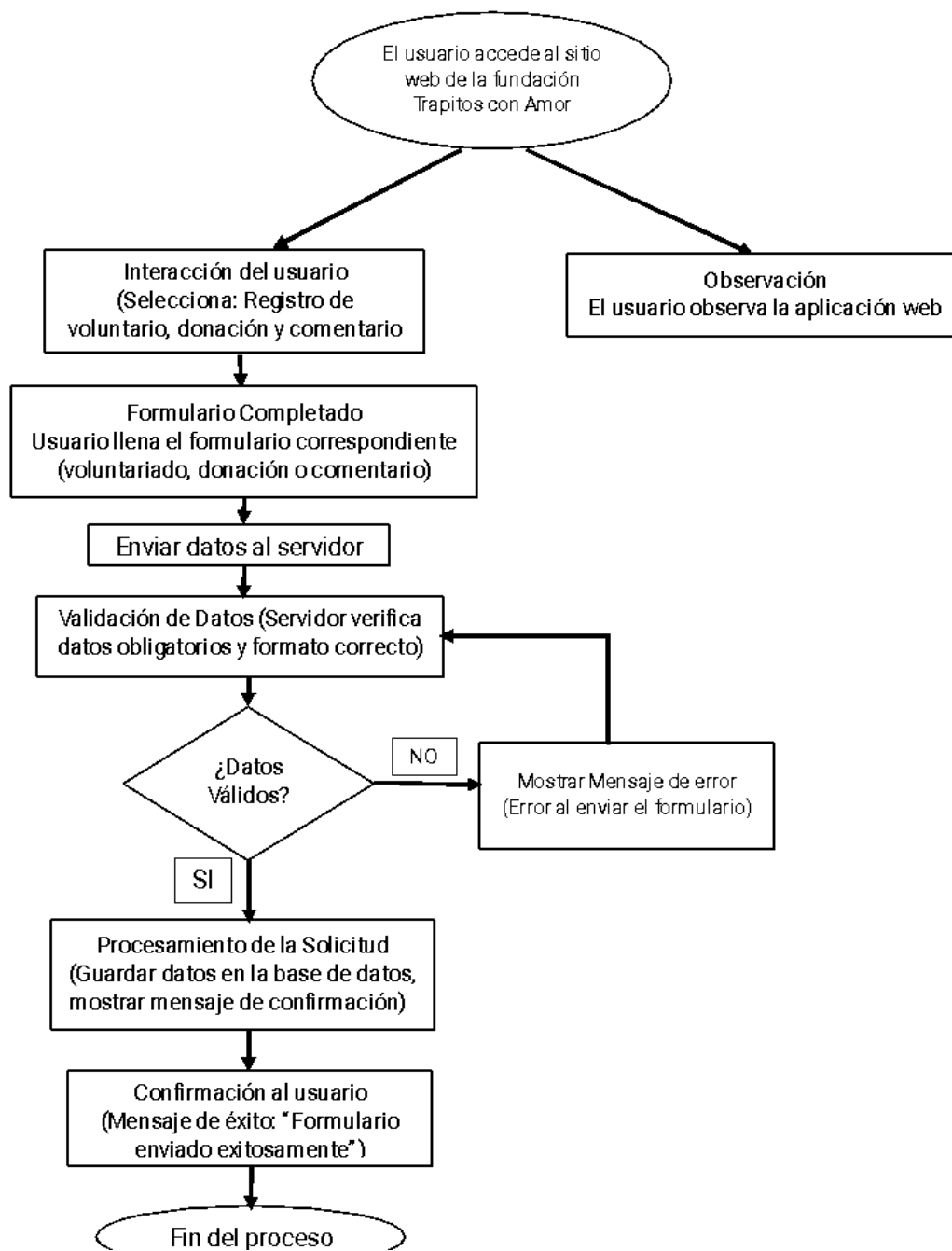


Diagrama de flujo de base de datos

Descripción:

Evidencia el funcionamiento de la base de datos del aplicativo web y el servidor que se utiliza.

Figura 6

Funcionamiento de la Base de datos

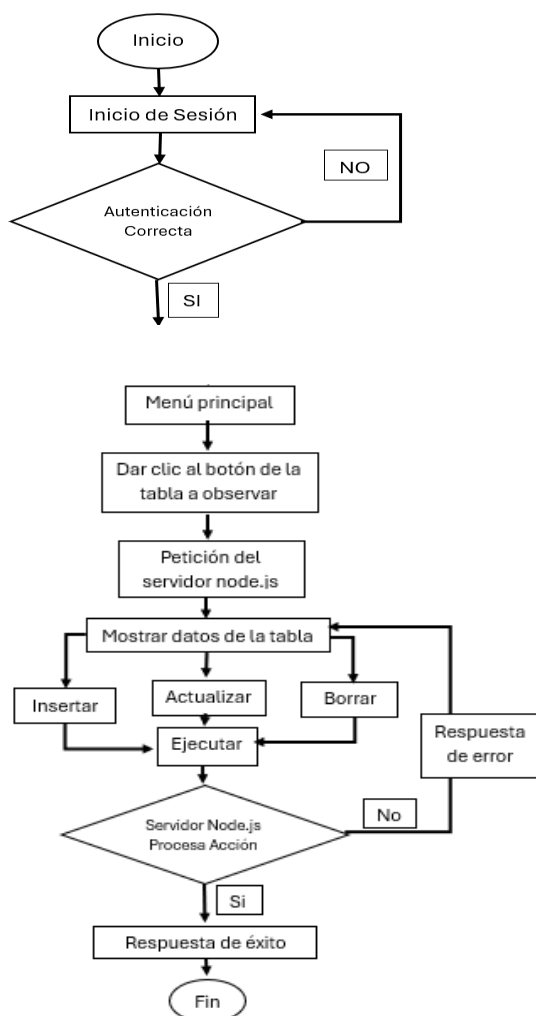


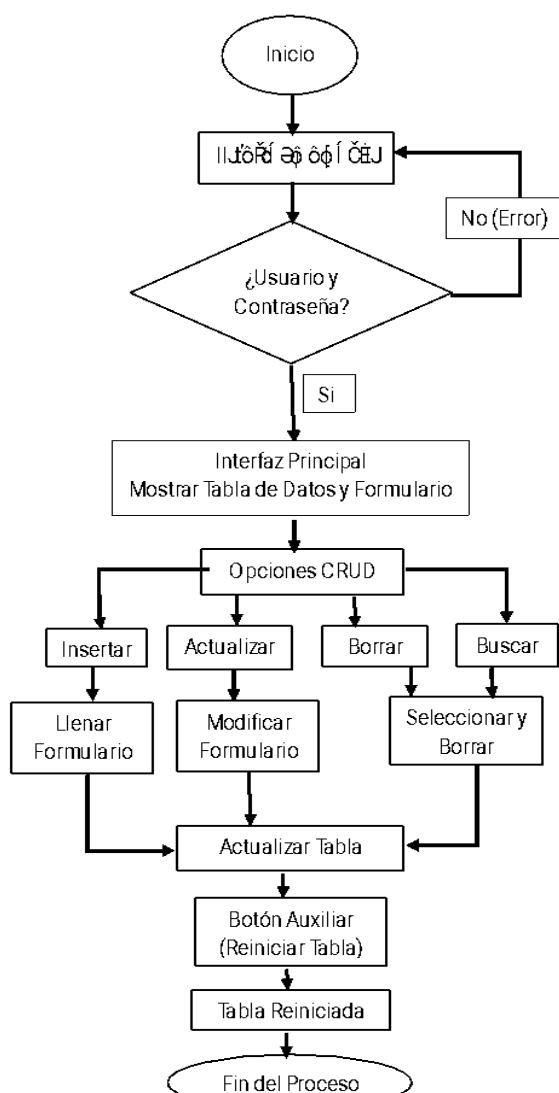
Diagrama de flujo de software

Descripción:

El manejo y uso del software ante los beneficiarios de la fundación puede evidenciarse.

Figura 7

Manejo y Uso del Software



Desarrollar el aplicativo web

Nota: En este punto se establecerá una breve descripción de cada código utilizado en las diferentes pestañas de la aplicación web, la base de datos y el software.

Análisis

Permitirá garantizar un rendimiento óptimo, una funcionalidad dinámica y seguridad basada en PHP y Node.js, lo cual ayuda a prevenir inyecciones y a mejorar la experiencia del usuario. Por ello, cada elemento ofrecido fue implementado cuidadosamente y diseñado con el propósito de dar a conocer la fundación, asegurando que sea fácil y rápida de utilizar.

“La implementación de código como PHP y Node.js en este proyecto fue cuidadosamente seleccionada para garantizar un rendimiento óptimo, una funcionalidad dinámica y una seguridad robusta. Cada uno de estos diseños aporta sus propias ventajas, haciendo que la plataforma sea eficiente, rápida y segura para los usuarios, lo que es esencial para lograr una experiencia de usuario positiva y fluida. La implementación de tecnologías como PHP y Node.js en este proyecto fue cuidadosamente seleccionada para garantizar un rendimiento óptimo, una funcionalidad dinámica y una seguridad robusta. Cada una de estas tecnologías aporta sus propias ventajas, haciendo que la plataforma sea eficiente, rápida y segura para los usuarios, lo que es esencial para lograr una experiencia de usuario positiva y fluida. En términos de rendimiento, PHP y Node.js ofrecen mecanismos complementarios para manejar grandes volúmenes de tráfico de manera eficiente. PHP es ideal para aplicaciones web de carga rápida, ya que su estructura de ejecución permite gestionar solicitudes HTTP de manera eficaz y se integra fácilmente con bases de datos como MySQL, que es clave para la gestión de datos de la fundación. La optimización del código PHP, junto con prácticas como la caché de datos y la minimización de consultas repetitivas a la base de datos, asegura que las páginas se carguen de

manera ágil. La seguridad fue una de las principales preocupaciones durante el desarrollo, dado que el sistema maneja información sensible, como datos personales de voluntarios y usuarios. PHP, junto con Node.js, ofrece un conjunto de herramientas que ayudan a mitigar las vulnerabilidades más comunes en aplicaciones web”- Welling, L., & Thompson, L. (2016). PHP and MySQL Web Development. Addison-Wesley Professional.

Programación (Cambiar en el cronograma)

Definición de drivers

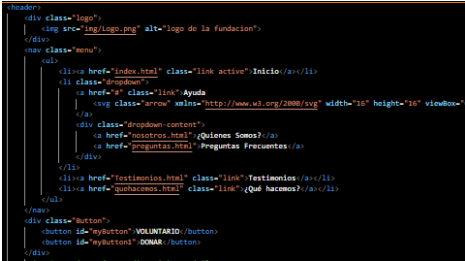
Requisitos Funcionales

Menú principal

Explicación del menú principal de cada pestaña que use la aplicación web.

Tabla 3

Menú principal

Código:	Descripción:
Html:  <pre> <header> <div class="logo"> </div> <div class="menu"> <li class="dropdown"> Inicio <li class="link">Ayuda <div class="arrow" xmins="http://www.v2.org/2008/svg" width="16" height="16" viewbox="0 0 16 16"> <div class="dropdown-content"> Quiénes Somos? Preguntas Frecuentes </div> <li class="link">Testimonios <li class="link">¿Qué hacemos? </div> </header> <div class="button"> <button id="mybutton" type="button" value="VOLUNTARIO">VOLUNTARIO</button> <button id="mybutton" type="button" value="DONAR">DONAR</button> </div> </pre>	Html: Se incluye un div con la clase logo, que contiene una imagen. La imagen es el logo de la fundación y se carga desde la carpeta img mediante el archivo Logo.png. El atributo alt proporciona una descripción alternativa en caso de que la imagen no pueda cargarse. El elemento <nav> agrupa el menú
Javascript:	

```
document.getElementById("myButton").onclick = function () {
  location.href = "Voluntario.html";
};
document.getElementById("myButton1").onclick = function () {
  location.href = "donar.html";
};
document.getElementById("myButton4").onclick = function () {
  location.href = "quehacemos.html";
};
```

Style:

```
/*Encabezado*/
header {
  display: flex;
  height: 60px;
  justify-content: space-between;
  align-items: center;
  padding: 60px;
  background: linear-gradient(90deg, rgba(255, 241, 194) 1
}

.logo {
  flex: 1;
  display: flex;
  align-items: center;
}

.logo img {
  height: 210px;
}

.menu {
  flex: 2;
  display: flex;
  justify-content: center;
  margin-right: 5px; /* Agrega un margen derecho al menú */
}

.menu ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
}

.menu li {
  margin: 0 20px;
  position: relative;
}

.menu a {
  text-decoration: none;
  color: black;
  font-size: 20px;
  display: flex;
  align-items: center;
  position: relative;
}
```

de navegación. `` es una lista desordenada que contiene los enlaces del menú. Cada elemento de lista `` contiene un enlace `<a>`, el cual, al hacer clic en el botón "Inicio", lleva a `index.html` u otra página según el nombre del botón.

Javascript: Este código permite,

mediante

`document.getElementById("myButton")`,

buscar en el código HTML un elemento con

el atributo `id="myButton"`. Utilizando

`.onclick`, se asigna una función que se

ejecutará cuando se haga clic en ese botón.

La función `function () {location.href =`

`"Voluntario.html";}` define la acción que

redirige a la página **Voluntario.html**

utilizando `location.href`, el cual cambia la

URL de la página actual.

Style: Permitirá llevar a cabo el ajuste

y diseño del menú para que sea visible en el

sitio donde se desea ubicar. Además, permitió

```

.menu .link:hover, .menu .active {
  transform: scale(1.1); /* Aumenta el tamaño del botón al 110% */
  border-radius: 8px; /* Aplica un borde redondeado al botón */
  transition: transform 0.3s ease, border-radius 0.3s ease; /* Agrega una t
}

/* Estilos para el icono de la flecha */
.arrow {
  margin-left: 0px; /* Ajusta según sea necesario */
  width: 20px; /* Tamaño ajustado */
  height: 20px; /* Tamaño ajustado */
  display: inline-block;
  position: relative;
  top: 2px; /* Ajuste para centrar verticalmente */
}

/* Estilos para el menú desplegable */
.dropdown {
  position: relative;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #25688C;
  min-width: 160px;
  box-shadow: 0 8px 0 #000;
  z-index: 1;
}

.dropdown-content a {
  color: white;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: justify; /* Justificar el texto dentro de los enlaces */
}

.dropdown-content a:nth-child(2) {
  white-space: nowrap; /* Evitar que el texto se divida en varias líneas */
}

.dropdown-content a:hover {
  background-color: #778899;
}

```

realizar un ajuste para que, al tocar cada botón, este se agrande utilizando transform.

```

.dropdown-content a:hover {
  background-color: #778899;
}

.dropdown:hover .dropdown-content {
  display: block;
}

button {
  color: white;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
  font-size: 16px;
  border-radius: 10px;
  transition: .3s ease;
  display: inline-flex;
  background-color: #25688C;
}

#myButton {
  margin-right: 12px; /* Separación de los b
}

.Button {
  flex: 1;
  display: flex;
  justify-content: flex-end;
  align-items: center;
  /*separa los botones del menu*/
  margin-left: 5px; /* Agrega un margen izqu
}

#myButton, #myButton1 {
  margin-left: 10px;
}

#myButton4{
  background-color: #878BD3;
  margin-top: 20px;
  color: black;
  font-size: 20px; /* Tamaño de la fuente */
  font-weight: bold;
}

```

Página Principal

En este punto se explicará el desarrollo de la página principal de la aplicación web, mostrando un pantallazo del código.

Portada #1

Tabla 4

portada#1

Código:	Descripción:
<p>Html:</p> <pre><section class="portada"> <div class="texto"> <h1>AYÚDANOS A AYUDAR</h1> </div> <div class="imagen"></div> </section></pre> <p>Style:</p> <pre>/* Sección de texto */ .portada .texto { flex: 1; /* Ocupa el 50% del ancho */ display: flex; justify-content: center; align-items: center; padding: 10px; /* Ajusta el padding según sea necesario */ background-color: rgba(255, 255, 255, 0.9); /* Fondo semitransparente */ } .portada .texto h1 { font-size: var(--hl-font-size, 4em); /* Usa una variable CSS para el tamaño de fuente */ color: black; text-align: center; /* Centra el texto horizontalmente */ align-items: center; line-height: 1.2; text-shadow: 2px 2px 0px black; /* Delineado de las letras */ transition: color 0.5s, font-size 0.5s; /* Transición suave para el cambio de color */ position: absolute; top: 60%; /* Alinea desde la mitad superior del contenedor */ transform: translate(-50%); /* Centra verticalmente con desplazamiento negativo */ } /* Sección de imagen */ .portada .imagen { flex: 1; /* Ocupa el 50% del ancho */ background-image: url("../img/grupo5.jpg"); /* Ruta relativa a la carpeta 'img' */ background-size: cover; /* Asegura que la imagen cubra todo el área */ background-position: center; /* Centra la imagen */ background-repeat: no-repeat; /* No repetir la imagen */ padding: 10px; /* Ajusta el padding según sea necesario */ }</pre> <p>Javascript:</p> <pre>// Función para cambiar el texto de la portada principal const h1ElementPrincipal = document.querySelector('.portada h1'); const coloresBandera = ['#FFDA44', '#0033A0', '#CE1126']; let currentIndex = 0; function cambiarColorYTamanoPrincipal() { h1ElementPrincipal.style.color = coloresBandera[currentIndex]; currentIndex = (currentIndex + 1) % coloresBandera.length; const nuevoTamano = h1ElementPrincipal.style.fontSize === '4em' ? '3em' : '4em'; h1ElementPrincipal.style.fontSize = nuevoTamano; h1ElementPrincipal.style.top = '60%'; h1ElementPrincipal.style.transform = 'translate(-50%)'; } cambiarColorYTamanoPrincipal(); setInterval(cambiarColorYTamanoPrincipal, 2000);</pre>	<p>Html: “La etiqueta <section> es un contenedor semántico utilizado para agrupar contenido relacionado. En este caso, todo lo que esté dentro de esta sección probablemente forma parte de una "portada" o una parte destacada de la página web”- HTML: HyperText Markup Language" - Referencia de HTML en MDN.</p> <p>Style: Permite colocar la imagen y ajustar su ancho o largo, así como el texto, según las preferencias del cliente.</p> <p>Javascript: Se muestra const coloresBandera = ['#FFDA44', '#0033A0', '#CE1126'], lo que permite definir los tres</p>

colores que se utilizarán de forma transitoria. La función `function cambiarColorYTamañoPrincipal()` se encarga de cambiar el color, el tamaño y la posición del elemento `<h1>`.

Portada #2

Tabla 5

Portada #2

Código:	Descripción:
<p>Html:</p> <pre><section class="portada1"> <div class="texto"> <h1>¡Gracias por no botar lo que se puede utilizar! -Trapitos con Amor</h1> </div> </section></pre>	<p>Html: “La etiqueta <code><section></code> es un contenedor semántico utilizado para agrupar contenido relacionado. En este caso, todo lo que esté dentro de esta sección probablemente forma parte de una "portada" o una parte destacada de la página web”- HTML: HyperText Markup Language" - Referencia de HTML en MDN.</p>
<p>Javascript:</p> <pre>const h1ElementPortada1 = document.querySelector('.portada1 .texto h1'); const frases = ["La falta de amor es la mayor pobreza - Teresa de Calcuta", "El que no vive para servir, no sirve para vivir - Teresa de Calcuta", "Si no se vive para los demás, la vida carece de sentido - Teresa de Calcuta", "¡Gracias por no botar lo que se puede utilizar! - Trapitos con Amor"]; let index = 0; function cambiarTextoPortada1() { h1ElementPortada1.textContent = frases[index]; index = (index + 1) % frases.length; } setInterval(cambiarTextoPortada1, 5000);</pre>	
<p>Style:</p> <pre>.portada1 { display: flex; height: 500px; /* Altura completa de la ventana */ position: relative; margin-top: 2.480px; /*me permite despegar de algo que este en la parte de arriba*/ } .portada1 .texto { display: flex; justify-content: center; align-items: center; padding: 30px; /* Ajusta el padding según sea necesario */ background: linear-gradient(90deg, rgba(255, 241, 104) 10%, rgba(255, 0, 0, 0.8) 60%); background-color: aliceblue; background: linear-gradient(90deg, rgba(255, 241, 104, 0.8) 0%, rgba(255, 241, 104, 0.8) 30%, rgba(255, 108, 236, 0.8) 40%, rgba(255, 108, 236, 0.8) 70%, rgba(234, 71, 71, 0.8) 75%, rgba(234, 71, 71, 0.8) 100%); }</pre>	<p>Javascript: Primero se utiliza <code>const</code> para definir las frases que se mostrarán. A continuación, se implementa la función</p>

encargada de realizar el cambio de texto, y, por último, se especifica el tiempo durante el cual se desea que se realice el cambio.

Style: En el estilo se establece para indicar el color de fondo, el tamaño de la fuente, la altura, el ancho y altura de la ventana donde estará el contenido.

Información de actividades

Tabla 6

Información de actividades

Código:	Descripción:
<p>Html:</p> <pre><div class="carousel"> <div class="anterior botones"> </div> <div class="imagenes"> </div> <div class="siguiente botones"> </div> </div></pre> <p>JavaScript:</p>	<p>Html: El contenido principal del body presenta un carrusel implementado mediante una división <div> que contiene imágenes insertadas con . En</p> <p>JavaScript: Se utilizó el evento DOMContentLoaded, el cual se dispara cuando el documento HTML ha sido completamente cargado y procesado por el navegador, pero</p>

```

document.addEventListener('DOMContentLoaded', function() {
  // Función para manejar el carrusel de imágenes
  const imagenes = ['img/imagen3.jpg', 'img/imagen1.png', 'img/imagen2.jpeg'];
  let cont = 0;

  function carrusel(contenedor) {
    contenedor.addEventListener('click', e => {
      const atras = contenedor.querySelector('#anterior');
      const adelante = contenedor.querySelector('#adelante');
      const img = contenedor.querySelector('.img');
      const tgt = e.target;

      if (tgt === atras) {
        cont = (cont > 0) ? cont - 1 : imagenes.length - 1;
        img.src = imagenes[cont];
      } else if (tgt === adelante) {
        cont = (cont < imagenes.length - 1) ? cont + 1 : 0;
        img.src = imagenes[cont];
      }
    });

    // Función para cambiar la imagen automáticamente
    function cambiarImagen() {
      cont = (cont < imagenes.length - 1) ? cont + 1 : 0;
      contenedor.querySelector('.img').src = imagenes[cont];
    }

    // Cambiar imagen cada 5 segundos
    setInterval(cambiarImagen, 5000);
  }

  // Inicialización del carrusel
  const contenedorCarrusel = document.querySelector('.carousel');
  carrusel(contenedorCarrusel);
}

```

Style:

```

.carousel{
  display:flex;
  align-items:center;
  justify-content:center;
  width:1070px;
  height:55vh;
  margin:0 auto; /*Centra horizontalmente*/
  position:relative; /* Posicionamiento absoluto para centrar dentro del main */
  left:50%; /* Posicionon desde el centro del main */
  transform:translateX(-50%); /* Ajustar el carrusel hacia la izquierda por la mitad de su propio ancho */
  top:26px; /*Ajustar la distancia desde la parte superior del main */
}
.botones{
  background-color: rgba(0,0,0,.2);
  position:absolute;
  top:50%;
  color: #fff;
  transform: translateY(-50%);
  box-shadow: 0 0 10px rgba(255, 255, 255, .2);
  transition:all 0.3s ease;
  padding:10px;
  cursor:pointer;
  z-index:1;
}
.anterior {
  left: 2%;
}
.siguiente {
  right: 2%;
}
.botones img:hover {
  transform: scale(1.2);
}

```

antes de que los recursos externos, como imágenes o estilos, hayan terminado de cargarse. Mediante una sentencia if-else, se permite que el usuario avance o retroceda a través de las imágenes utilizando las flechas, o, en caso de no realizar ninguna acción, las imágenes se cambian automáticamente cada 5 segundos. “**const imagenes =**

['img/imagen3.jpg', 'img/imagen1.png', 'img/imagen2.jpeg']; let cont = 0;

Aquí se define un array imágenes que contiene las rutas de las imágenes que serán mostradas en el carrusel. La variable cont se usa para llevar el seguimiento de la imagen actual mostrada. **function carrusel(contenedor) {...}** Esta función recibe como parámetro un contenedor que tiene los elementos del carrusel. Dentro de esta función

```

.imagenes{
  display:flex;
  flex-direction:column;
  align-items:center;
  justify-content:center;
}
.img{
  box-shadow:0px 2px 6px 0px rgba(0,0,0,0.50);
  height:350px;
  width:990px; /*990px*/
  border-radius:3px; /*bordes de la imagen*/
  margin:0px;
  object-fit:cover;
  filter:saturate(90%); /*iluminar la imagen*/
}

```

se gestiona la lógica del carrusel, tanto para los cambios manuales, realizados al hacer clic, como para los cambios automáticos, que ocurren cada 5 segundos.

```

contenedor.addEventListener('click', e => {
  const atras = contenedor.querySelector('#anterior');
  const adelante = contenedor.querySelector('#adelante');
  const img = contenedor.querySelector('.img');
  const tgt = e.target;

  if (tgt === atras) {
    cont = (cont > 0) ? cont - 1 : imagenes.length - 1;
    img.src = imagenes[cont];
  } else if (tgt === adelante) {
    cont = (cont < imagenes.length - 1) ? cont + 1 : 0;
    img.src = imagenes[cont];
  }
});

```

El código agrega un **eventListener** al contenedor para detectar clics. Luego, según dónde se haga clic (en el botón de "anterior" o "adelante"), se actualiza la imagen mostrada:

Si se hace clic en el botón "anterior", se reduce el contador `cont` para mostrar la imagen anterior. Si `cont` llega a 0, se reinicia al índice de la última imagen. Si se hace clic en el botón "adelante", se incrementa el contador `cont` para mostrar la siguiente imagen. Si `cont`

llega al final de la lista de imágenes, se reinicia al índice de la primera imagen”- JavaScript: The Definitive Guide - David Flanagan 2020.

Style En el archivo de estilos, se realizaron ajustes para que el carrusel tenga el diseño deseado.

Detalles y carrusel extranjeros

Tabla 7

Detalles y Carrusel

Código:	Descripción:
Detalles	
<p>Html:</p> <pre><div class="detalles"> <div class="contenido"> <div class="texto"> <h2>DETALLES DE LA FUNDACIÓN</h2> <p>Recibimos y entregamos donaciones de todo tipo nuevo y usado, Eje <button id="myButton4">¿Qué hacemos?</button> </div> </div> </div></pre>	<p>Html: Se muestra que el código esta dividido en tres <div> en el que se nombran mediante class, se hace el uso del , <p>, <h2> y <button>, para mostrar el texto, la imagen y un botón nombrado ¿Qué hacemos?</p> <p>Style: Las clases se llaman usando un punto seguido del nombre</p>

```

.img{
  box-shadow:0px 2px 6px 0px rgba(0,0,0,0.50);
  height:350px;
  width:990px; /*990px*/
  border-radius:3px;/*bordes de la imagen*/
  margin:0px;
  object-fit:cover;
  filter: saturate(90%);/*iluminar la imagen*/
}

.detalles {
  margin-top: 60px;
}

.contenido {
  display: flex;
  align-items: center; /* Alinear verticalmente el contenido */
  padding-left: 15px;/*despejar de la pared izquierda*/
}

.foto {
  height: 510px;
  width: 500px;
  margin-right: 20px; /* Espacio entre la imagen y el texto */
}

```

```

.texto {
  display: flex;
  flex-direction: column;
  align-items: center; /* Centrar el contenido horizontalmente */
  flex: 1; /* Hacer que el texto ocupe el espacio restante */
  font-size: 16px; /* Tamaño de fuente base */
}

.texto h2 {
  margin: 0 0 10px 0; /* Quitar el margen del titulo */
  text-align: center; /* Centrar el texto del titulo */
  font-size: 24px; /* Tamaño de fuente más grande para el titulo */
}

.texto p {
  margin: 0; /* Quitar el margen del párrafo */
  text-align: left; /* Alinear el texto del párrafo a la izquierda */
  text-align: justify; /* Justificar el texto del párrafo */
  font-size: 27px; /* Aumentar el tamaño de la fuente del párrafo */
  line-height: 1.5; /* Espacio entre líneas para mejor legibilidad */
}

```

de la clase asociada al <div>. A través de controladores, se define la dirección en la que se posicionan los elementos hijos en cada clase. Por ejemplo, width se usa para ajustar el ancho del contenido, y border-radius para redondear los bordes de las imágenes, entre otros. Estos estilos permiten que la imagen se alinee a la derecha y el texto a la izquierda.

Carrusel Extranjeros

Html:

```

<div class="carousel2">
  <h2 class="extran">VOLUNTARIOS EXTRANJEROS</h2>
  <button class="prev">&#10094;</button>
  <div class="carousel2-container">
    <div class="carousel2-slide">
      
      
      
      
      
    </div>
  </div>
  <button class="next">&#10095;</button>
</div>

```

Style:

Html: En el Contenedor

principal del carrusel se presenta el <div class="carousel2">, que agrupa todo el contenido relacionado, permitiendo mostrar un título con <h2></h2>, también se evidencia mediante un <button></button> para navegar la imagen hacia al lado

```

/*Carusel 2*/
.carousel2 {
  position: relative;
  max-width: 1000px;
  margin: auto;
  overflow: hidden;
  margin-top: 15px;
}

.carousel2-container {
  display: flex;
  justify-content: center;
  align-items: center;
}

.carousel2-slide {
  display: flex;
  transition: transform 0.5s ease-in-out;
}

.carousel2-slide img {
  width: 300px;
  height: 220px;
  margin: 0 15px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

```

```

.prev, .next {
  position: absolute;
  top: 63%;
  transform: translateY(-50%);
  background-color: rgba(0, 0, 0, 0.5);
  color: white;
  border: none;
  padding: 10px;
  cursor: pointer;
  z-index: 10; /* Asegúrate de que los botones estén por encima de las imágenes */
}

.prev {
  left: 0;
}

.next {
  right: 0;
}

```

JavaScript:

izquierdo o derecho mostrando en la aplicación diseñada los símbolos en estilo flecha (◀) (▶), en el código html se indica mediante el `❮` y `❯`. Para mostrar las imágenes que forman parte del carrusel se usó ``, cada una con su respectiva fuente (src) y texto alternativo (alt). **Style:** Se utiliza la propiedad `position: relative;`, que permite posicionar elementos dentro del contenedor de forma relativa. Esto resulta útil para los botones de navegación. Por otro lado, la propiedad `overflow: hidden;` oculta cualquier contenido que se desborde del contenedor, como imágenes que se encuentren fuera de la vista principal del carrusel.-

Duckett, J. (2011). **HTML and CSS: Design and Build Websites.** Wiley. (margin-top: 15px) Añade un margen superior de 15 píxeles para

```

document.addEventListener('DOMContentLoaded', function() {
  const carousels = document.querySelectorAll('.carousel2');

  carousels.forEach((carousel) => {
    const prev = carousel.querySelector('.prev');
    const next = carousel.querySelector('.next');
    const carouselSlide = carousel.querySelector('.carousel2-slide');
    const images = carousel.querySelectorAll('.carousel2-slide img');

    let counter = 0;
    const size = images[0].clientWidth + 30; // 300px width + 30px margin

    function updateCarousel() {
      carouselSlide.style.transform = `translateX(${ -size * counter }px)`;
      prev.style.display = counter === 0 ? 'none' : 'block';
      next.style.display = counter >= images.length - 3 ? 'none' : 'block';
      console.log("Updated carousel, counter: ", counter); // Añadir para depuración
    }

    next.addEventListener('click', () => {
      if (counter >= images.length - 3) return; // stop at last set of images
      carouselSlide.style.transition = "transform 0.4s ease-in-out";
      counter++;
      console.log("Next clicked, counter: ", counter); // Añadir para depuración
      updateCarousel();
    });

    prev.addEventListener('click', () => {
      if (counter <= 0) return; // stop at first set of images
      carouselSlide.style.transition = "transform 0.4s ease-in-out";
      counter--;
      console.log("Prev clicked, counter: ", counter); // Añadir para depuración
      updateCarousel();
    });

    // Inicializa el estado del carrusel
    updateCarousel();
  });
});

```

separar el carrusel del contenido superior. (**box-shadow: 0 4px 8px rgba (0, 0, 0, 0.1) ;**) Añade una sombra a las imágenes para darles un efecto de profundidad.

Javascript:

“(DOMContentLoaded) El evento se dispara cuando el contenido HTML ha sido completamente cargado y analizado. Esto asegura que el código se ejecute solo después de que la estructura HTML esté lista.

(document.querySelectorAll(‘. carousel2’)) Selecciona todos los elementos en el documento que tienen la clase carousel2. Esto es útil si tienes múltiples carruseles en la misma página. (**prev** y **next**) Son los botones que permiten navegar hacia adelante y hacia atrás en el carrusel. (**carouselSlide**) Es el contenedor que tiene las imágenes del carrusel. Este elemento se moverá horizontalmente

para mostrar diferentes imágenes.

(images): Selecciona todas las imágenes dentro del contenedor **(carouselSlide)**. **(next)** Al hacer clic, incrementa el contador para avanzar el carrusel, y luego llama a

(updateCarousel) para mover las imágenes. **(Prev)** Al hacer clic, disminuye el contador para retroceder el carrusel, y luego llama a **(updateCarousel)**, **(console.log())**

Se utilizan para imprimir el valor del contador cada vez que se actualiza el carrusel o cuando se hace clic en los botones **(prev y next)**. Esto es útil para depurar y asegurarse de que el carrusel esté funcionando

correctamente”- **W3Schools**.

"JavaScript Tutorial". Recurso educativo en línea para aprender desarrollo web. Refsnes Data.

Comentarios

Tabla 8

Comentarios

Código:	Descripción:
<p>Html:</p> <pre data-bbox="203 611 673 886"> <div class="comment-section"> <h2>Deja Tú Comentario</h2> <form id="commentForm" class="commentForm"> <div id="nameField" class="hidden"> <label for="username">Tu nombre:</label> <input type="text" id="username" name="username"> </div> <label for="commentType">Selecciona el tipo de comentario:</label> <select id="commentType" name="commentType"> <option value="anonimo">Anónimo</option> <option value="nombre">Diligencia Nombre</option> </select> <label for="comment">Comentario:</label> <textarea id="comment" name="comment" rows="4"></textarea> <button type="submit">Enviar Comentario</button> </form> <div id="commentList"> <!-- Aquí se mostrarán los comentarios --> </div> <div id="errorMessage" class="error-message"></div> </div> </pre>	<p>Html: En el contenedor se presenta el <div>, el label está vinculado al campo de entrada de texto para el nombre mediante el atributo, el label está vinculado (select) que permite al usuario elegir si quiere hacer un comentario anónimo o con nombre, también permitirá que cuando los campos se llenen, al hacer clic en el botón "Enviar Comentario", el formulario será procesado y el comentario será mostrado dentro de #commentList.</p>
<p>Javascript:</p> <pre data-bbox="203 995 673 1570"> document.addEventListener('DOMContentLoaded', function() { const commentId = Date.now(); const serverUrl = 'http://localhost/Fundación/php/comments.php'; const errorMessage = document.getElementById('errorMessage'); const usernameInput = document.getElementById('username'); const commentInput = document.getElementById('comment'); document.getElementById('commentForm').addEventListener('submit', function(event) { event.preventDefault(); var username = usernameInput.value.trim(); var commentType = document.getElementById('commentType').value; var comment = commentInput.value.trim(); // Resetear mensaje de error errorMessage.textContent = ''; // Validar que los campos requeridos no estén vacíos if (username === '' && commentType === 'nombre') { errorMessage.textContent = 'Por favor, ingresa tu nombre.'; return; // Detener la ejecución de la función } if (comment === '') { errorMessage.textContent = 'Por favor, ingresa tu comentario.'; return; // Detener la ejecución de la función } // Validar que el nombre no contenga caracteres especiales if (commentType === 'nombre' && containsSpecialCharacters(username)) { errorMessage.textContent = 'Por favor, ingresa un nombre válido.'; return; // Detener la ejecución de la función } // Validar que el nombre no contenga palabras no deseadas if (commentType === 'nombre' && containsUnwantedWords(username)) { errorMessage.textContent = 'Por favor, ingresa un nombre válido.'; return; // Detener la ejecución de la función } }); } </pre>	<p>Javascript: “Date.now () devuelve la cantidad de milisegundos transcurridos desde el 1 de enero de 1970 (también conocido como "Unix Epoch"). Se usa para generar un identificador único basado en el tiempo actual. En este contexto, se utiliza como un identificador para el comentario o para el usuario que lo está enviando”- Eich, Brendan</p>

```

var newComment = {
  commenterId: commenterId,
  username: commentType === 'nombre' ? username : '',
  comment: comment
};

fetch(serverUrl, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(newComment)
})
.then(response => response.json())
.then(data => {
  .then(data.id => {
    if (data.id) {
      errorMessage.textContent = 'Comentario enviado y en revisión.';
      // Limpiar los campos después de enviar el comentario
      usernameInput.value = '';
      commentInput.value = '';
    } else if (data.error) {
      errorMessage.textContent = 'Error: ' + data.error;
    }
  })
})
.catch(error => {
  console.error('Error:', error);
  errorMessage.textContent = 'Error al enviar el comentario.';
});

```

```

// función para verificar si el nombre contiene caracteres especiales
function containsSpecialCharacters(name) {
  var regex = /[@!$%&'()*+,-./:;<=>?`{|}~]/; // Definir la expresión regular para caracteres especiales
  return regex.test(name);
}

// función para verificar si el nombre contiene palabras no deseadas
function containsUnwantedWords(name) {
  // Lista de palabras no deseadas (puedes ajustar según tus necesidades)
  var unwantedWords = ['moco', 'puma', 'salchicha', 'marica', 'ladrones'];
  // Convertir el nombre a minúsculas para hacer la comparación insensible a mayúsculas
  var lowerCaseName = name.toLowerCase();
  // Verificar si el nombre contiene alguna palabra no deseada
  for (var i = 0; i < unwantedWords.length; i++) {
    if (lowerCaseName.includes(unwantedWords[i])) {
      return true;
    }
  }
  return false;
}

function showComments() {
  fetch(serverUrl)
  .then(response => response.json())
  .then(comments => {
    document.getElementById('commentlist').innerHTML = '';
    comments.forEach(comment => {
      var commentElement = document.createElement('div');
      commentElement.classList.add('comment');

      var commenterInfo = document.createElement('p');
      commenterInfo.textContent = comment.username ? 'Nombre: ' + comment.username : 'Anónimo';

      var commentText = document.createElement('p');
      commentText.textContent = comment.comment;

      var deleteButton = document.createElement('span');
      deleteButton.classList.add('delete-btn');
      deleteButton.textContent = 'Eliminar';
      deleteButton.addEventListener('click', function() {
        deleteComment(comment.id);
      });

      commentElement.appendChild(commenterInfo);
      commentElement.appendChild(commentText);
      commentElement.appendChild(deleteButton);

      document.getElementById('commentlist').appendChild(commentElement);
    });
  });
}

```

```

function deleteComment(commentId) {
  fetch(`${serverUrl}?id=${commentId}`, {
    method: 'DELETE'
  })
  .then(response => response.json())
  .then(data => {
    showComments();
    adjustFooterPadding(); // Ajustar el padding-bottom después de eliminar comentario
  });
}

showComments();

```

Php:

Comments.php:

(1995).

<http://localhost/Fundación/php/comments.php>

se encuentra allí porque es la url del servidor por medio de php. Mediante este bloque maneja el evento de submit del formulario. Se previene el comportamiento por defecto con `event.preventDefault()` para que la página no se recargue al enviar el formulario, permitiendo la visualización de comentarios en una página web, usándolo para la interacción del usuario mediante la selección de un tipo de comentario, el envío de comentarios y su validación, así como la eliminación de comentarios.

Php: Comments.php: En la primera

línea se incluye el archivo `db.php`, que contiene las funciones necesarias para conectar a la base de datos MySQL. En los códigos header establecen los encabezados HTTP necesarios para manejar solicitudes AJAX desde cualquier origen (*) y permiten métodos HTTP POST, GET, DELETE y OPTIONS. El sistema también realiza

```

<?php
include 'db.php';

header('Content-Type: application/json');
header('Access-Control-Allow-Origin: *'); // Cambia * por el dominio permitido en producción
header('Access-Control-Allow-Methods: POST, GET, DELETE, OPTIONS');
header('Access-Control-Allow-Headers: Content-Type');

$method = $_SERVER['REQUEST_METHOD'];

switch ($method) {
    case 'GET':
        $sql = "SELECT * FROM comentarios WHERE published = 1"; // Solo obtener comentarios publicados
        $stmt = $conn->prepare($sql);
        $stmt->execute();
        $result = $stmt->get_result();
        $comments = [];
        while ($row = $result->fetch_assoc()) {
            $comments[] = $row;
        }
        echo json_encode($comments);
        $stmt->close();
        break;

    case 'POST':
        $data = json_decode(file_get_contents("php://input"), true);
        if (!isset($data['commenterId'], $data['username'], $data['comment'])) {
            echo json_encode(["error" => "Faltan datos en la solicitud."]);
            exit();
        }
        $commenterId = filter_var($data['commenterId'], FILTER_SANITIZE_NUMBER_INT);
        $username = filter_var($data['username'], FILTER_SANITIZE_STRING);
        $comment = filter_var($data['comment'], FILTER_SANITIZE_STRING);

        if (empty($commenterId) || empty($username) || empty($comment)) {
            echo json_encode(["error" => "Datos inválidos."]);
            exit();
        }

        $sql = "INSERT INTO comentarios (commenterId, username, comment, published) VALUES (?, ?, ?, 0)";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param('iss', $commenterId, $username, $comment);

        if ($stmt->execute()) {
            echo json_encode(["id" => $conn->insert_id]);
        }

```

```

        if ($stmt->execute()) {
            echo json_encode(["id" => $conn->insert_id]);
        } else {
            echo json_encode(["error" => $stmt->error]);
        }
        $stmt->close();
        break;

    case 'DELETE':
        if (!isset($_GET['id'])) {
            echo json_encode(["error" => "ID no especificado."]);
            exit();
        }

        $id = filter_var($_GET['id'], FILTER_SANITIZE_NUMBER_INT);

        if (empty($id)) {
            echo json_encode(["error" => "ID inválido."]);
            exit();
        }

        $sql = "DELETE FROM comentarios WHERE id = ?";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param('i', $id);

        if ($stmt->execute()) {
            echo json_encode(["success" => true]);
        } else {
            echo json_encode(["error" => $stmt->error]);
        }
        $stmt->close();
        break;

    case 'OPTIONS':
        // Manejar solicitudes OPTIONS para CORS preflight
        http_response_code(200);
        break;

    default:
        echo json_encode(["error" => "Método no permitido."]);
        http_response_code(405);
        break;
}

```

solicitudes GET para obtener comentarios,

maneja solicitudes OPTIONS y utiliza

declaraciones preparadas (prepared

statements) para evitar inyecciones SQL.

Además, lleva a cabo validaciones básicas de

entrada antes de procesar cada tipo de

solicitud.

En el archivo db.php, la variable

\$servername hace referencia al nombre del

host donde se encuentra la base de datos. En

este caso, "127.0.0.1" representa la dirección

IP utilizada. local, lo que sugiere que se está

utilizando una base de datos local. \$username:

El nombre de usuario para acceder a la base

de datos. \$password: La contraseña para

acceder a la base de datos. Aquí está vacío, lo

que podría ser un error de seguridad.

\$dbname: El nombre de la base de datos a la

que se quiere conectarse"- Percona. (s.f).

Este código permite realizar una conexión con

la base de datos MYSQL mediante el servidor

XAMPP.

```

    $sql = "DELETE FROM comentarios WHERE id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param('i', $id);

    if ($stmt->execute()) {
        | echo json_encode(["success" => true]);
    } else {
        | echo json_encode(["error" => $stmt->error]);
    }
    $stmt->close();
    break;

case 'OPTIONS':
    // Manejar solicitudes OPTIONS para CORS preflight
    http_response_code(200);
    break;

default:
    echo json_encode(["error" => "Método no permitido."]);
    http_response_code(405);
    break;
}

$conn->close();
?>

```

Style: Con este se pudo manejar el diseño del ancho y alto del formulario. Además, permitió ubicar los campos en los lugares deseados dentro de la pestaña.

db.php:

```

<?php
$servername = "127.0.0.1";
$username = "root";
$password = "";
$dbname = "voluntariodb";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>

```

Style:

```

.comment-section {
    width: 90%;
    max-width: 900px;
    margin: 20px auto; /* Ajuste para centrar verticalmente y dar margen */
    padding: 20px;
    border: 1px solid black;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    box-sizing: border-box;
    background-color: white;
    margin-top: 38px;
}

.comment {
    position: relative;
    margin-bottom: 18px;
    padding: 10px;
    border: 1px solid black /*#eee*/;
    border-radius: 5px;
}

.comment .delete-btn {
    position: absolute;
    top: 5px;
    right: 5px;
    cursor: pointer;
    color: red;
}

```

```

.comment .delete-btn {
  position: absolute;
  top: 5px;
  right: 5px;
  cursor: pointer;
  color: red;
}

.hidden {
  display: none;
}

.comment-section h2 {
  text-align: center; /* Justificar el texto del h2 */
}

.comment-section label {
  display: block;
  margin-bottom: 5px;
}

.comment-section input[type="text"],
.comment-section textarea,
.comment-section select,
.comment-section button {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border: 1px solid black; /* #ccc */
  border-radius: 5px;
  box-sizing: border-box;
}

```

```

.comment-section button {
  display: block; /* Asegura que el botón sea un bloque */
  width: auto; /* Ajusta el ancho automáticamente */
  margin: left; /* Centra horizontalmente el botón */
  background-color: #007bff;
  color: white;
  border: none;
  cursor: pointer;
  padding: 10px;
  border-radius: 5px;
  text-align: center; /* Centrar el texto del botón */
}

.comment-section button:hover {
  background-color: #0056b3;
}

.commentForm {
  margin-top: 20px; /* permite que el contenido se separe del título */
}

.error-message {
  color: red; /* Cambia el color del texto a rojo */
  /* Otros estilos opcionales */
  font-weight: bold; /* Puedes hacer el texto en negrita */
  /* Ajusta otros estilos según sea necesario */
  margin-top: 5px;
}

```

Integración con Redes Sociales

Tabla 9

Integración

Código:

Descripción:

```

/*inserto publicidad en el cuerpo de la aplicación*/
.video-youtube {
  position: relative;
  width: 95%; /*El ancho del video*/
  padding-bottom: 45%; /* Relación de aspecto 16:9 */
  height: 28vh; /*El alto del video*/
  overflow: hidden;
  max-width: 1200px; /*se debe ajustar el video segun el ancho y alto*/
  margin: auto; /* Centrar el video */
}

.video-youtube iframe {
  position: absolute;
  top: 0;
  left: 12;
  width: 100%;
  height: 100%;
  border: 0;
  justify-content: center;
}

/*Video facebook*/

.video-facebook {
  position: relative;
  width: 95%; /* Ajusta el ancho del contenedor del video */
  max-width: 1200px; /* Ancho máximo del contenedor del video */
  margin: 10px auto;
  margin-bottom: -10px; /*pone espacio entre videos*/
}

.video-facebook iframe {
  width: 100%;
  height: auto;
  border: 0;
  display: block; /* Asegura que el iframe se ajuste correctamente */
}

```

¿Quiénes Somos?

Descripción:

En este se ve el desarrollo de cada botón del menú desplegable que se encuentra en esta pestaña.

Historia, misión, visión, Fundadores y Colaboradores

Tabla 10

¿Quiénes Somos?

Código:	Descripción:
Html:	Se establece que la pestaña Fundadores y colaboradores es la que va a tener la información de cada persona y foto.

```

<section class="portada">
  <h1>¡Quéén Somos!</h1>
</section>
<main>
  <div id="main-container">
    <div id="menu" class="menu-info">
      <div id="menu-toggle">
        <div class="btn-menu"></div>
        <div class="btn-menu"></div>
        <div class="btn-menu"></div>
      </div>
      <div id="profile">
        
      </div>
      <ul id="menu-items">
        <li><a href="historia.html"><i class="fas fa-file-alt"></i><span>Historia</span></li>
        <li><a href="mision.html"><i class="fas fa-flag"></i><span>Misión</span></li>
        <li><a href="vision.html"><i class="fas fa-eye"></i><span>Visión</span></li>
        <li><a href="fundadores.html"><i class="fas fa-user"></i><span>Fundadores</span></li>
        <li><a href="colaboradores.html"><i class="fas fa-users"></i><span>Colaboradores</span></li>
        <li><a href="pensamientos.html"><i class="fas fa-brain"></i><span>Pensamientos</span></li>
      </ul>
    </div>
    <div id="para"><p>Sé agradable con las personas. Tal vez serás menospreciado, rechazado o ignorado, pero dife
  </div>

```

```

</div>
<div id="para"><p>Sé agradable con las personas. Tal vez serás menospreciado, rechazado o
</div>
<div id="content">
  Contenido Misión
</div>

```

```

<footer>
  <div class="column1">
    <div class="redes">
      <ul class="list-unstyled">
        <li><a href="https://www.facebook.com/profile.php?id=100074473202487" class="linkedin" target="blank"></li>
        <li><a href="https://www.instagram.com/trapiitosconamor2020?igsh=8a440280029a18b6=" class="instagram" ta
        <li><a href="#" class="tiktok" target="blank"><i class="fa-brands fa-tiktok"></i></li>
        <li><a href="#" class="youtube" target="blank"><i class="fa-brands fa-youtube"></i></li>
      </ul>
    </div>
    <div class="cont-footer">
      <ul class="list-unstyled">
        <li><a href="#">Privacidad y Políticas</li>
        <li><a href="#">Términos y Condiciones</li>
      </ul>
    </div>
    <div class="column2">
      <p>©copy; 2024 | Fundación Trapiitos Con Amor | Todos los derechos reservados</p>
    </div>
  </div>
</footer>

```

Javascript:

```

document.getElementById("myButton").onclick = function () {
  location.href = "voluntario.html";
};
document.getElementById("myButton1").onclick = function () {
  location.href = "donar.html";
};

//me permite que oculte lo que no quiero ver cuando se contraiga.
document.addEventListener("DOMContentLoaded", function() {
  var menuToggle = document.getElementById("menu-toggle");
  var menu = document.getElementById("menu");
  var paragraph = document.getElementById("para");

  menuToggle.addEventListener("click", function() {
    menu.classList.toggle("collapsed");
    paragraph.classList.toggle("collapsed");
  });
});

// Selecciona el elemento h1
const h1Element = document.querySelector('.portada h1');

// Cambia el color de fondo y el tamaño de la fuente
h1Element.style.setProperty('--h1-font-size', '4em');
// Arreglo con los colores de la bandera de Colombia
const coloresBandera = ['#FFD444', '#003366', '#E11261'];
let currentIndex = 0;
// Función para cambiar el color del texto
function cambiarColorTexto() {
  h1Element.style.color = coloresBandera[currentIndex];
  currentIndex = (currentIndex + 1) % coloresBandera.length;
}
// Cambia el color inicialmente y establece un intervalo para cambiar cada 2 segundos
cambiarColorTexto();
setInterval(cambiarColorTexto, 2000); // Cambia cada 2 segundos (2000 milisegundos)

```

```

// Función para alternar la visibilidad del menú desplegable
// Función para alternar la clase de menú abierto/cerrado
// Función para alternar la clase de menú abierto/cerrado
function toggleMenu() {
  var mobileMenu = document.querySelector('.mobile-menu');
  mobileMenu.classList.toggle('open'); // Alterna la clase 'open' en el menú móvil
}

```

Style:

HTML: Se encontrará una sección en

la parte superior de la pestaña en la que el texto se define por un h1, que permite indicar la introducción a la página. El main contiene el contenido principal de la pestaña; mediante un div se establece un div que permite colocar un menú en la parte lateral con opciones de enlaces en estilo botón. Por consiguiente, se tiene otro div, el que permite colocarle al menú lateral un botón tipo hamburguesa.

También, dentro de este se encuentra otro div el que se define la img, que permite visualizar el favicon de la empresa. Se puede ver un uso de ul, li, a y i, que permite hacer la lista de elementos que se quiere que contenga el menú. Con otro div se puede mostrar un texto usando p; después de ello se puede ver un div para mostrar el contenido que se mostrará cuando se le dé clic en alguno de los botones que se definieron.

Javascript: En este código se puede

ver en la primera parte una redirección al hacer clic en botones.

Style: Permite definir los colores y diseño para las páginas de Historia, misión, visión, Fundadores y Colaboradores; también funciona para reubicar los botones, el texto y los iconos de una manera visible y organizada, entre otros.

Preguntas Frecuentes

Descripción:

En este se podrá ver el desarrollo de las preguntas que al dar clic en la flecha de despliegue muestra la respuesta.

Menú desplegable

Tabla 11

Menú desplegable

Código:	Descripción:
Html:	Html: En este código se puede ver un <code></header></code> en el que, como primera parte, un <code><section></code> permite mostrar la portada. En la segunda parte, dentro del <code><main></code> , se encuentra una secuencia de

```

document.getElementById("myButton").onclick = function () {
  location.href = "Voluntario.html";
};
document.getElementById("myButton1").onclick = function () {
  location.href = "donar.html";
};
document.addEventListener("DOMContentLoaded", function() {
  var botones = document.querySelectorAll(".btnshow");

  botones.forEach(function(boton) {
    boton.addEventListener("click", function() {
      var respuesta = this.nextElementSibling;
      respuesta.classList.toggle("mostrar");
    });
  });
});

const h1Element = document.querySelector('.portada h1');

// Cambia el color de fondo y el tamaño de la fuente
h1Element.style.setProperty('--h1-font-size', '4em');
// Arreglo con los colores de la bandera de Colombia
const coloresBandera = ['#FFD444', '#0033A0', '#CE1126'];
let currentIndex = 0;
// Función para cambiar el color del texto
function cambiarColorTexto() {
  h1Element.style.color = coloresBandera[currentIndex];
  currentIndex = (currentIndex + 1) % coloresBandera.length;
}

// Cambia el color inicialmente y establece un intervalo para cambiar cada 2 segundos
cambiarColorTexto();
setInterval(cambiarColorTexto, 2000); // Cambia cada 2 segundos (2000 milisegundos)

function toggleMenu() {
  const menu = document.querySelector('.mobile-menu');
  menu.classList.toggle('open');
}

```

Style:

```

.video-youtube {
  margin: auto; /* Centran el video */
}

.video-youtube iframe {
  position: absolute;
  top: 0;
  left: 12;
  width: 100%;
  height: 100%;
  border: 0;
  justify-content: center;
}

/*Video facebook*/

.video-facebook {
  position: relative;
  width: 95%; /* Ajusta el ancho del contenedor del video */
  max-width: 1200px; /* Ancho máximo del contenedor del video */
  margin: 10px auto;
  margin-bottom: -10px; /*pone espacio entre videos*/
}

.video-facebook iframe {
  width: 100%;
  height: auto;
  border: 0;
  display: block; /* Asegura que el iframe se ajuste correctamente */
}

h4 {
  margin-left: 5px; /*despeja el texto de la columna*/
}

```

color y tamaño del texto del título y permite cambio de color del texto del título con los colores de la bandera de Colombia utilizando **const**, **let**, **function** y **setInterval**.

Style: Permite definir los colores y

diseño para la página, también funciona para reubicar los botones, el texto y los iconos de una manera visible y organizada, entre otros.

Testimonios

Descripción:

En este se puede evidenciar el desarrollo de la pestaña testimonios que se muestran mediante cuadrados y dentro de cada uno de ellos una imagen.

Tabla 12

Testimonios

Código:	Descripción:
Html:  	Html: Se tiene una sección portada en la que está definida por <code><h1></code> y un <code><section></code> , Tiene un main con contenedor de cuadrícula (<code><div class="grid-container"></code>) que organiza diversos elementos de contenido en una cuadrícula, Incluye imágenes y iframes para videos de YouTube y Facebook, permitiendo mostrar contenido multimedia.
Javascript: 	Javascript: Se encuentra una redirección al dar clic en botones usando el código: <pre>document.getElementById("").onclick = function () { location.href = ""; };</pre> Se evidencia un cambio dinámico del color del texto en un elemento, en este caso el índice se reinicia al final del arreglo para crear un ciclo continuo, además, se

Style:

```

@keyframes slide {
  0% {
    transform: translateX(0);
  }
  100% {
    transform: translateX(100%);
  }
}

menu {
  position: absolute;
  top: 10px;
  left: 10px;
  background-color: #333;
  color: white;
  padding: 5px;
  border-radius: 5px;
  width: 100px;
  z-index: 1000;
}

menu ul {
  list-style-type: none;
  padding: 0;
  margin: 0;
}

menu ul li {
  padding: 5px 0;
}

menu ul li a {
  color: white;
  text-decoration: none;
}

menu ul li a:hover {
  background-color: #555;
}

body {
  font-family: sans-serif;
  font-size: 16px;
  background-color: #f0f0f0;
  padding: 20px;
}

h1 {
  text-align: center;
  margin: 0;
}

p {
  text-align: center;
  margin: 10px 0;
}

a {
  color: #007bff;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

img {
  max-width: 100%;
  height: auto;
}

pre {
  background-color: #2e3436;
  color: #eeeeec;
  padding: 10px;
  border: 1px solid #2e3436;
}

```

llama la función inicialmente para aplicar el primer color y luego se utiliza setInterval para cambiar el color del texto cada 2 segundos (2000 milisegundos). La función para activar/desactivar un menú móvil, Esta función selecciona un elemento con la clase mobile-menu.

“con menu.classList.toggle('open'), se añade o remueve la clase open cuando se activa la función. Suele usarse para mostrar u ocultar un menú en dispositivos móviles, dependiendo del estilo CSS asociado a la clase open” Mozilla Developer Network (MDN) Web Docs. (2024).

Style: Permite definir los colores y diseño para la página de Testimonios; también funciona para reubicar los botones, el texto y los iconos de una manera visible y organizada, entre otros. Este código permite poner la cantidad de columnas que se quieren y un ancho con

que se ajusta de fr (**grid-template-columns: repeat(3, 1fr);**). Me ajusta las imágenes y videos dentro de cada cuadro.


¿Qué hacemos?

Descripción:

En este se verá un código, en el que se explica cómo se realizó el proceso de subtítulos en el que tiene varios cuadrillos dependiendo de la cantidad de imágenes que se quieren mostrar.

Tabla 13

¿Qué hacemos?

Código:	Descripción:
<p>Html:</p> 	<p>Html: En este se encontrarán subtítulos de nivel 1 (<h1>) que muestra el texto, Un contenedor de tipo div con la clase "grid-container", esta clase es usada con estilos CSS para organizar su contenido en una cuadrícula (grid). Cada uno de estos div contiene una imagen que se muestra en la cuadrícula. Las imágenes tienen una ruta de archivo, lo que sugiere que están almacenadas localmente en una carpeta.</p>
<p>Javascript:</p>	

```

js testimonios.js > ...
document.getElementById("myButton").onclick = function () {
  location.href = "Voluntario.html";
};
document.getElementById("myButton1").onclick = function () {
  location.href = "donar.html";
};
const hElement = document.querySelector('.portada h1');

// Cambia el color de fondo y el tamaño de la fuente
hElement.style.setProperty('--h1-font-size', '4em');
// Arreglo con los colores de la bandera de Colombia
const coloresBandera = ['#FFD444', '#0033A0', '#CE1126'];
let currentIndex = 0;
// Función para cambiar el color del texto
function cambiarColorTexto() {
  hElement.style.color = coloresBandera[currentIndex];
  currentIndex = (currentIndex + 1) % coloresBandera.length;
}

// Cambia el color inicialmente y establece un intervalo para cambiar cada 2
cambiarColorTexto();
setInterval(cambiarColorTexto, 2000); // Cambia cada 2 segundos (2000 miliseg)

function toggleMenu() {
  const menu = document.querySelector('.mobile-menu');
  menu.classList.toggle('open');
}

```

Style:

```

<div class="mobile-menu">
  <ul>
    <li>Inicio</li>
    <li>Servicios</li>
    <li>Contacto</li>
  </ul>
</div>

```

Javascript: En esta se encuentra

una redirección al dar clic en botones usando el código:

```
document.getElementById("").onclick = function () { location.href = ""; };
```

Se evidencia un cambio dinámico

del color del texto en un elemento, en este

caso el índice se reinicia al final del arreglo

para crear un ciclo continuo, además, se

llama la función inicialmente para aplicar

el primer color y luego se utiliza setInterval

para cambiar el color del texto cada 2

segundos (2000 milisegundos). La función

para activar/desactivar un menú móvil,

Esta función selecciona un elemento con la

clase mobile-menu.

“con menu.classList.toggle('open'),

se añade o remueve la clase open cuando

se activa la función. Esto suele usarse para

mostrar u ocultar un menú en dispositivos

móviles, dependiendo del estilo CSS

asociado a la clase open” Mozilla

Developer Network (MDN) Web Docs.
(2024).

Style: Permite definir los colores y diseño para la página de Testimonios; también funciona para reubicar los botones, el texto y los iconos de una manera visible y organizada, entre otros. Este código permite poner la cantidad de columnas que se quieren y un ancho con que se ajusta de fr (**grid-template-columns: repeat(3, 1fr);**). Me ajusta las imágenes y videos dentro de cada cuadro, y permite color los subtítulos.

Donar

Descripción:

Este código permitio hacer la construcción y diseño de un formulario que permitirá a los usuarios donar en la fundación.

Tabla 14

Donar

Código:	Descripción:
---------	--------------

HTML:

```

2 <h1 lang="es">
3 </h1>
4 <div>
5 <div id="content">
6 <input type="text" value="Nombre" />
7 <input type="text" value="Apellido" />
8 <input type="text" value="Correo" />
9 <input type="text" value="Telefono" />
10 <input type="text" value="Direccion" />
11 <input type="text" value="Calle" />
12 <input type="text" value="Ciudad" />
13 <input type="text" value="País" />
14 <input type="text" value="Código Postal" />
15 <input type="text" value="Fecha de Nacimiento" />
16 <input type="text" value="Sexo" />
17 <input type="text" value="Profesión" />
18 <input type="text" value="Ocupación" />
19 <input type="text" value="Ocupación" />
20 <input type="text" value="Ocupación" />
21 <input type="text" value="Ocupación" />
22 <input type="text" value="Ocupación" />
23 <input type="text" value="Ocupación" />
24 <input type="text" value="Ocupación" />
25 <input type="text" value="Ocupación" />
26 <input type="text" value="Ocupación" />
27 <input type="text" value="Ocupación" />
28 <input type="text" value="Ocupación" />
29 <input type="text" value="Ocupación" />
30 <input type="text" value="Ocupación" />
31 <input type="text" value="Ocupación" />
32 <input type="text" value="Ocupación" />
33 <input type="text" value="Ocupación" />
34 <input type="text" value="Ocupación" />
35 <input type="text" value="Ocupación" />
36 <input type="text" value="Ocupación" />
37 <input type="text" value="Ocupación" />
38 <input type="text" value="Ocupación" />
39 <input type="text" value="Ocupación" />
40 <input type="text" value="Ocupación" />
41 <input type="text" value="Ocupación" />
42 <input type="text" value="Ocupación" />
43 <input type="text" value="Ocupación" />
44 <input type="text" value="Ocupación" />
45 <input type="text" value="Ocupación" />
46 <input type="text" value="Ocupación" />
47 <input type="text" value="Ocupación" />
48 <input type="text" value="Ocupación" />
49 <input type="text" value="Ocupación" />
50 <input type="text" value="Ocupación" />
51 <input type="text" value="Ocupación" />
52 <input type="text" value="Ocupación" />

```

JAVASCRIPT:

Server.

```

1 addEventListener('DOMContentLoaded', function() {
2   serverUrl = 'http://localhost:3000/submit';
3   errorMessage = document.getElementById('errorMessage');
4   thankYouMessage = document.getElementById('thankYouMessage');
5   select = document.getElementById('pais');
6   submitInput = document.getElementById('submit');
7
8   submitInput.addEventListener('click', function(event) {
9     event.preventDefault();
10
11     let username = submitInput.value.trim();
12     let pais = select.value;
13
14     // Validar mensaje de error
15     errorMessage.textContent = '';
16
17     // Validar que los campos requeridos no estén vacíos
18     if (username === '' || pais === '') {
19       errorMessage.textContent = 'Todos los campos son obligatorios.';
20       return; // Detener la ejecución de la función.
21     }
22
23     let data = new FormData();
24     data.append('pais', pais);
25     data.append('username', username);
26
27     fetch(serverUrl, {
28       method: 'POST',
29       body: data
30     })
31     .then(response => response.json())
32     .then(data => {
33       // Manejar la respuesta
34     });
35   });
36 });

```

Donar:

```

1 document.getElementById("myButton").onclick = function () {
2   location.href = "voluntario.html";
3 };
4 document.getElementById("myButton1").onclick = function () {
5   location.href = "donar.html";
6 };
7 const hElement = document.querySelector('.portada h1');
8
9 // Cambia el color de fondo y el tamaño de la fuente
10 hElement.style.setProperty('--hi-font-size', '4em');
11 // Arreglo con los colores de la bandera de Colombia
12 const coloresBandera = ['#FFD444', '#0033A0', '#CE1126'];
13 let currentIndex = 0;
14 // Función para cambiar el color del texto
15 function cambiarColorTexto() {
16   hElement.style.color = coloresBandera[currentIndex];
17   currentIndex = (currentIndex + 1) % coloresBandera.length;
18 }
19
20 // Cambia el color inicialmente y establece un intervalo para cambiar cada 2 segundos
21 cambiarColorTexto();
22 setInterval(cambiarColorTexto, 2000); // Cambia cada 2 segundos (2000 milisegundos)
23
24 function toggleMenu() {
25   const menu = document.querySelector('.mobile-menu');
26   menu.classList.toggle('open');
27 }

```

Style.

HTML: En este código se evidencia un

`<section>` en el que permite mostrar una portada, en la que se usa `<h1>` para ver el título según el nombre que se coloque. Se tiene un `<main>` que define la estructura principal del

contenido de la pestaña, se tiene un `div` en el que permite mostrar el formulario, por consiguiente, se tiene unos campos en el formulario en el que se hace uso de `label`, `select`, `option`. También, mediante el `div` permitirá mostrar el mensaje de error y aceptación.

JAVASCRIPT:

Server: En este código se puede

evidenciar que permite enviar los datos a un servidor mediante una solicitud `POST`, en la que se usa el código

`document.addEventListener('DOMContentLoaded', function() {...})`: que asegura que todo el contenido de la página esté completamente

cargado antes de ejecutar el código que maneja el formulario. Se tiene algunas declaraciones de variables, `serverUrl` permite definir la URL del servidor, `errorMessage` la que selecciona el

```

}
portada h1 {
font-size: 100px; font-weight: bold; /* Se usa un tamaño fijo para el tamaño de fuente */
border-radius: 50%; /* Añade bordes redondeados al fondo */
padding: 5px 10px; /* Ajusta el padding según sea necesario */
margin: 0px 0 10px 0; /* Añade un margen superior e inferior */
text-align: center; /* Centra horizontalmente el contenido */
align-items: center; /* Centra verticalmente el contenido */
text-align: center; /* Centra el texto horizontalmente */
display: flex;
line-height: 1.2;
text-shadow: 2px 2px 0px black; /* Delineado de las letras */
transition: color 0.5s; /* Transición suave para el cambio de color */
}

/* Estilos para el icono de la flecha */
arrow {
margin-left: 10px; /* Ajusta según sea necesario */
width: 20px; /* Tamaño ajustado */
height: 20px; /* Tamaño ajustado */
display: inline-block;
position: relative;
top: 20px; /* Ajusta para centrar verticalmente */
}

/* Estilos para el menú desactivado */
disabled {
position: relative;
}

```

elemento html con el id y el mensaje de error, **thankYouMessage** la que muestra el mensaje de agradecimiento, y para seleccionar el campo **paisSelect** y **usernameInput**.

Server:

Donar.php.

```

<?php
include 'dbdonar.php';

header('Content-Type: application/json');
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Methods: POST');
header('Access-Control-Allow-Headers: Content-Type');

// Verificar el método de solicitud
if ($_SERVER['REQUEST_METHOD'] != 'POST') {
    echo json_encode(["success" => false, "message" => "Método de solicitud no válido"]);
    exit();
}

// Obtener los datos de la solicitud
$data = $_POST;

$requiredFields = ['pais', 'username'];
foreach ($requiredFields as $field) {
    if (isset($data[$field]) || empty(trim($data[$field]))) {
        echo json_encode(["success" => false, "message" => "Faltan datos en la solicitud"]);
        exit();
    }
}

// Verificar que se recibieron todos los datos necesarios
$pais = filter_var($data['pais'], FILTER_SANITIZE_STRING);
$username = filter_var($data['username'], FILTER_SANITIZE_STRING);

// Preparar la consulta SQL
$sql = "INSERT INTO donar (pais, username) VALUES (?, ?)";
$stmt = $conn->prepare($sql);

```

dbdonar.php:

```

<?php
$servername = "127.0.0.1";
$username = "root";
$password = "";
$dbname = "donardb";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>

```

Donar.js: Se encuentra una redirección al dar clic en botones usando el código:

```
document.getElementById("").onclick = function () { location.href = ""; };
```

Se evidencia un cambio dinámico del color del texto en un elemento, en este caso el índice se reinicia al final del arreglo para crear un ciclo continuo, además, se llama la función inicialmente para aplicar el primer color y luego se utiliza setInterval para cambiar el color del texto cada 2 segundos (2000 milisegundos). La función para activar/desactivar un menú móvil, Esta función selecciona un elemento con la clase mobile-menu.

“con menu.classList.toggle('open'), se añade o remueve la clase open cuando se activa la función. Se usa para mostrar u ocultar un menú en dispositivos móviles, dependiendo del

estilo CSS asociado a la clase open” Mozilla Developer Network (MDN) Web Docs. (2024).

STYLE: Permite definir los colores, centrar los campos de mi formulario y demás información que se necesita para hacer un aspecto visible y creativo en la pantalla, se usó algunos códigos que permiten ajustar el ancho del formulario como flex, border-radius para poner más redondeado el botón o el campo del formulario, entre otros códigos que hicieron una pestaña con estilo donativo.

SERVER:

Donar.php:

Se evidencia un código con una solicitud POST, tenemos como primera parte el código para hacer el llamado al archivo de la conexión de la base de datos, como segunda parte se tiene la configuración de los encabezados que se establece la respuesta mediante JSON, permitiendo solicitudes desde cualquier origen (útil para manejar CORS). Se tiene la verificación de solicitud de tipo POST usando una sentencia If, se tiene una línea de código

`$data = $_POST;` que obtiene los datos enviados por el método POST. Se tiene un `foreach` y `$requiredFields` que verifica los campos requeridos y se tiene un `filter_var` que permite eliminar cualquier carácter potencialmente peligroso. También, se tiene el llamado a la tabla en la que se agrega los datos registrados en el formulario, consiguiendo que mediante una sentencia `if` permita validar la consulta y arroje el mensaje de error o aprobado el envío de los datos a la base de datos, para hacer el registro por medio de `$stmt` y por seguridad se usa el cierre de la conexión mediante `$stmt->close();` `$conn->close();`.

Dbdonar.php:

Se realiza la definición de las variables para la conexión de la base de datos, se establece la conexión `$conn` y un `if` que permite verificar sin la conexión fue exitosa o falló.

```

/* content */
#bandera {
  margin-left: 8px;
}

/* Ajustes para el contenido cuando el menú está colapsado */
#menu.collapsed ~ #content {
  transform: translateX(-52px); /* Ajusta el valor para mover el contenido a la izquierda */
  margin-top: -600px; /* Ajusta según sea necesario */
  padding: 20px; /* Ajusta según sea necesario */
  font-size: 1.7em; /* Aumenta el tamaño del texto */
  max-width: 1200px; /* Ajusta el ancho máximo según sea necesario */
  text-align: center; /* Centra el texto */
}

/* Ajustes para el formulario específico */
#menu.collapsed ~ #content .commentForm {
  margin-right: auto;
  margin-left: auto;
  max-width: 800px; /* Ajusta el ancho máximo del formulario */
}
#menu.collapsed ~ #content h2 {
  margin-left: 2px;
}

/* Resto de estilos... */

#menu-toggle .btn-menu {
  width: 30px;
  height: 30px;
  background-color: #255; /* Cambia el color del toggle a blanco */
  margin-bottom: 5px;
}

#menu-toggle .btn-menu:last-child {
  margin-bottom: 0; /* Elimina el margen inferior del último botón */
}

```

Que permite mostrar el nombre de cada palabra de la lista a la que se quiere poner el menú y también al dar clic a la palabra nos lleve al enlace correspondiente.

Style: Permite definir los colores que elegí para el color de fondo del menú desplegable y la letra; también permite reubicar los botones, el logo, el texto y los iconos de una manera visible y organizada, entre otros.

Javascript: Permite que cuando se contraiga el menú o despliegue el texto no dañe su diseño, como también el menú desplegable no pierda su arquitectura. Por eso se usa un manejo de menú y párrafo cuando se contrae, cuando la página ha sido cargada por DOMContentLoaded, después, que suceda esto al dar clic en el botón del menú se alterna la clase collapsed en ambos elementos en la que

permite ocultar o mostrar el contenido del menú.

Parte principal de los formularios

Descripción:

Se muestra la parte del contenido que está ubicado en la parte izquierda de la pestaña.

Tabla 16

Parte principal formularios

Código:	Descripción:
<p>Html:</p> <pre> <h1> </h1> <h2> </h2> <h3> </h3> <p> </p> </pre>	<p>Html: Este código permite introducir texto usando <code><h1></code> que este nos da un título grande, <code><h2></code> da un título mediano y <code><h3></code> un poco más pequeño y con el uso de la <code><p></code> permite usar el texto en estilo párrafo y de una manera visible.</p>
<p>Style:</p> <pre> #content h1{ margin-top: 5px; text-align: center; } #content h2{ margin-top: 5px; } #content h3{ margin-top: 5px; } #content p{ margin-top: 5px;/*se separa el titulo del parrafo*/ } </pre>	<p>Style: Este estilo permite tener una letra separada del título además de ello centrado, organizado y visible.</p>
<p>Javascript:</p>	<p>Javascript: Permite que cuando se contraiga el menú o despliegue el texto no dañe su diseño, como también el menú desplegable no pierda su arquitectura. Por</p>

```

document.getElementById("myButton").onclick = function () {
  location.href = "Voluntario.html";
};
//me permite que oculte lo que no quiero ver cuando se contraiga.
document.addEventListener("DOMContentLoaded", function() {
  var menuToggle = document.getElementById('menu-toggle');
  var menu = document.getElementById('menu');
  var paragraph = document.getElementById('parr');

  menuToggle.addEventListener('click', function() {
    menu.classList.toggle('collapsed');
    paragraph.classList.toggle('collapsed');
  });
});

// Selecciona el elemento h1
const h1Element = document.querySelector('.portada h1');

// Cambia el color de fondo y el tamaño de la fuente
h1Element.style.setProperty('--h1-font-size', '4em');
// Arreglo con los colores de la bandera de Colombia
const coloresBandera = ['#FFD444', '#003366', '#CE1126'];
let currentIndex = 0;
// Función para cambiar el color del texto
function cambiarColorTexto() {
  h1Element.style.color = coloresBandera[currentIndex];
  currentIndex = (currentIndex + 1) % coloresBandera.length;
}

// Cambia el color inicialmente y establece un intervalo para cambiar cada 2 segundos
cambiarColorTexto();
setInterval(cambiarColorTexto, 2000); // Cambia cada 2 segundos (2000 milise

// Función para alternar la visibilidad del menú desplegable
// Función para alternar la clase de menú abierto/cerrado
// Función para alternar la clase de menú abierto/cerrado
function toggleMenu() {
  var mobileMenu = document.querySelector('.mobile-menu');
  mobileMenu.classList.toggle('open'); // Alterna la clase 'open' en el menú
}

```

eso se usa un manejo de menú y párrafo cuando se contrae, cuando la página ha sido cargada por DOMContentLoaded, después, que suceda esto al dar clic en el botón del menú se alterna la clase collapsed en ambos elementos en la que permite ocultar o mostrar el contenido del menú.

Explicación del formulario

Tabla 17

Explicación del Formulario

Código:	Descripción:
Html:	Html: Los formularios se envuelve dentro de un div y el titulo está en un nivel 2 usando h2. En los elementos del formulario se presenta mediante etiquetas <label> y campos <input>, cada campo tiene el atributo required, lo que indica que el formulario no puede enviarse sin

```

<div id="content">
  <h2 class="form-title">Formulario de Otra Entidad</h2>
  <!-- Formulario de voluntariado altruista -->
  <form id="commentForm" class="commentForm">
    <div id="otraentidadid">
      <label for="username">Nombre y Apellido Estudiante:</label>
      <input id="username" name="username" required>
      <label for="telefono">Teléfono de contacto Estudiante :</label>
      <input id="telefono" name="telefono" required>
      <label for="departamento">Departamento Estudiante:</label>
      <input id="departamento" name="departamento" required>
      <label for="municipio">Municipio Estudiante:</label>
      <input id="municipio" name="municipio" required>
      <label for="entidad">Entidad:</label>
      <input id="entidad" name="entidad" required>
      <label for="contactoentidad">Contacto de la Entidad:</label>
      <input id="contactoentidad" name="contactoentidad" required>
      <label for="lugar">Lugar de la Entidad:</label>
      <input id="lugar" name="lugar" required>
      <label for="direccion">Dirección de la Entidad:</label>
      <input id="direccion" name="direccion" required>
      <label for="otraentidaddias">Cuántos días o meses:</label>
      <input id="otraentidaddias" name="otraentidaddias" required>
      <label for="otraentidadmotivo">(Por qué quiere ser voluntario en la fundación):</label>
      <textarea id="otraentidadmotivo" name="otraentidadmotivo" rows="4" required></textarea>
      <label for="otraentidadactividad">(Qué actividad va a realizar en la fundación):</label>
      <textarea id="otraentidadactividad" name="otraentidadactividad" rows="4" required></textarea>
    </div>
    <button type="submit">Enviar Comentario</button>
    <div id="errorMessage" style="color: red;></div>
  </form>
</div>

```

Style:

```

.dash {
  display: inline-block;
  width: 18px; /* Ancho ajustable según necesidad */
  height: 2px; /* Altura ajustable según necesidad */
  background-color: black; /* Color ajustable */
}
//Bandera//
#bandera {
  margin-left: 8px;
}
/* Ajustes para el contenido cuando el menú está colapsado */
#menu.collapsed ~ #content {
  transform: translateX(-52px); /* Ajusta el valor para mover el contenido a la izquierda */
  margin-top: -60px; /* Ajusta según sea necesario */
  padding: 20px; /* Ajusta según sea necesario */
  font-size: 1.7em; /* Aumenta el tamaño del texto */
  max-width: 1200px; /* Ajusta el ancho máximo según sea necesario */
  text-align: center; /* Centra el texto */
}
/* Ajustes para el formulario específico */
#menu.collapsed ~ #content .commentForm {
  margin-right: auto;
  margin-left: auto;
  max-width: 800px; /* Ajusta el ancho máximo del formulario */
}
#menu.collapsed ~ #content h2 {
  margin-left: 2px;
}

```

Javascript:

1.

que estos campos sean completados, se tiene un botón para el envío que se usó button y otro mensaje error mediante un div.

Style: Este estilo permite ajustar

los campos del formulario y su marco, de modo que los elementos queden alineados dentro de él.

Además, permite que el título y el menú se adapten, añadiendo color y estilo.

Javascript:

1. Un evento

document.addEventListener

ner('DOMContentLoaded', function() {...})

que asegura que el código de la función no se ejecute hasta que todo el contenido

HTML este completamente cargado. Se

define de variables con const para

almacenar referencias a los elementos del

formulario como los campos de entrada, la

recopilación de valores se obtiene con

input.value.trim() y la función trim()

elimina espacios en blanco al principio y al

```

document.getElementById('contacto').addEventListener('submit', function(event) {
  // Validar campos
  const nombre = document.getElementById('nombre');
  const telefono = document.getElementById('telefono');
  const direccion = document.getElementById('direccion');
  const email = document.getElementById('email');
  const actividad = document.getElementById('actividad');
  const departamento = document.getElementById('departamento');
  const municipio = document.getElementById('municipio');
  const contacto = document.getElementById('contacto');
  const direccion = document.getElementById('direccion');
  const email = document.getElementById('email');
  const actividad = document.getElementById('actividad');

  // Validar que los campos requeridos no estén vacíos
  if (nombre.value === '' || telefono.value === '' || direccion.value === '' || email.value === '' || actividad.value === '' || departamento.value === '' || municipio.value === '' || contacto.value === '') {
    // Mensaje de error
    errorMessage.textContent = '...';
    return; // Detiene la ejecución de la función
  }

  // Crear objeto de datos
  const data = {
    nombre: nombre.value,
    telefono: telefono.value,
    direccion: direccion.value,
    email: email.value,
    actividad: actividad.value,
    departamento: departamento.value,
    municipio: municipio.value,
    contacto: contacto.value
  };

  // Enviar datos al servidor
  fetch('contacto.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  })
  .then(response => response.json())
  .then(data => {
    // Si data.success es true
    if (data.success) {
      // Mensaje de éxito
      errorMessage.textContent = '¡Felicitaciones!';
      // Limpiar los campos después de enviar el formulario
      document.getElementById('contacto').reset();
    } else {
      // Mensaje de error
      errorMessage.textContent = 'Error: ' + data.message;
    }
  })
  .catch(error => {
    // Mensaje de error
    errorMessage.textContent = 'Error: ' + error;
  });
});

```

final del texto. Con la validación de campos permite validar si un campo está vacío me muestra un mensaje de error `errorMessage.textContent = '...'` y se detiene la ejecución de la función `return;`. Con el `formData` que permite crear un objeto que almacena los valores del formulario, se usa `fetch()` para hacer una solicitud HTTP al servidor.

2.

```

// Selecciona el elemento h1
const h1Element = document.querySelector('.portada h1');

// Cambia el color de fondo y el tamaño de la fuente
h1Element.style.setProperty('--h1-font-size', '4em');
// Arreglo con los colores de la bandera de Colombia
const coloresBandera = ['#FFD444', '#003366', '#CE1126'];
let currentIndex = 0;
// Función para cambiar el color del texto
function cambiarColorTexto() {
  h1Element.style.color = coloresBandera[currentIndex];
  currentIndex = (currentIndex + 1) % coloresBandera.length;
}

// Cambia el color inicialmente y establece un intervalo para cambiar cada 2 segundos
cambiarColorTexto();
setInterval(cambiarColorTexto, 2000); // Cambia cada 2 segundos (2000 milisegundos)

// Función para alternar la visibilidad del menú desplegable
// Función para alternar la clase de menú abierto/cerrado
// Función para alternar la clase de menú abierto/cerrado
function toggleMenu() {
  var mobileMenu = document.querySelector('.mobile-menu');
  mobileMenu.classList.toggle('open'); // Alterna la clase 'open' en el menú
}

```

2. En esta se encuentra

una redirección al dar clic en botones usando el código: `document.getElementById('').onclick = function () { location.href = ''; };`

Se evidencia un cambio

Php:

1.

```

<?php
$servername = "127.0.0.1";
$username = "root";
$password = "";
$dbname = "voluntariodb";

// Crear conexión
$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar conexión
if ($conn->connect_error) {
  die("Conexión fallida: " . $conn->connect_error);
}
?>

```

dinámico del color del texto en un elemento, en este caso el índice se reinicia al final del arreglo para crear un ciclo continuo, además, se llama la función inicialmente para aplicar el primer color y luego se utiliza `setInterval` para cambiar el color del texto cada 2 segundos (2000

2.

```

@stranidad.php
<?php
include 'db_connect.php';

// Configurar los encabezados para JSON y CORS
header('Content-Type: application/json');
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Methods: POST');
header('Access-Control-Allow-Headers: Content-Type');

// Verificar el método de solicitud
if ($_SERVER['REQUEST_METHOD'] != 'POST') {
    echo json_encode(['success' => false, 'message' => 'Método de solicitud no válido']);
    exit();
}

// Obtener los datos de la solicitud
$data = $_POST;

// Verificar que se recibieron todos los datos necesarios
$requeridos = ['username', 'telefono', 'departamento', 'municipio', 'entidad', 'contactoentidad', 'lugar'];
foreach ($requeridos as $field) {
    if (isset($data[$field]) || empty(trim($data[$field]))) {
        echo json_encode(['success' => false, 'message' => 'Faltan datos en la solicitud.']);
        exit();
    }
}

// Sanitizar y validar datos
$username = filter_var($data['username'], FILTER_SANITIZE_STRING);
$telefono = filter_var($data['telefono'], FILTER_SANITIZE_STRING);
$departamento = filter_var($data['departamento'], FILTER_SANITIZE_STRING);
$municipio = filter_var($data['municipio'], FILTER_SANITIZE_STRING);
$entidad = filter_var($data['entidad'], FILTER_SANITIZE_STRING);
$contactoentidad = filter_var($data['contactoentidad'], FILTER_SANITIZE_STRING);
$lugar = filter_var($data['lugar'], FILTER_SANITIZE_STRING);
$direccion = filter_var($data['direccion'], FILTER_SANITIZE_STRING);
$ciudad = filter_var($data['ciudad'], FILTER_SANITIZE_STRING);
$activo = filter_var($data['otraentidadactiva'], FILTER_SANITIZE_STRING);
$actividad = filter_var($data['otraentidadactividad'], FILTER_SANITIZE_STRING);

// Preparar la consulta SQL
$sql = "INSERT INTO estudiantes_otra_entidad (username_estudiante, telefono_estudiante, departamento_estudiante,
$stmt = $conn->prepare($sql);

if ($stmt == false) {
    echo json_encode(['success' => false, 'message' => 'Error al preparar la consulta: ' . $conn->error]);
    exit();
}

$stmt->bind_param('ssssssssss', $username, $telefono, $departamento, $municipio, $entidad, $contactoentidad, $lugar, $direccion, $ciudad, $activo, $actividad);

if ($stmt->execute()) {
    echo json_encode(['success' => true, 'message' => 'Formulario enviado exitosamente.']);
} else {
    echo json_encode(['success' => false, 'message' => 'Error al guardar la información: ' . $stmt->error]);
}

$stmt->close();
$conn->close();
?>

```

milisegundos). La función para activar/desactivar un menú móvil, Esta función selecciona un elemento con la clase mobile-menu.

“con

menu.classList.toggle('open'), se añade o remueve la clase open cuando se activa la función. Esto suele usarse para mostrar u ocultar un menú en dispositivos móviles, dependiendo del estilo CSS asociado a la clase open” Mozilla Developer Network (MDN) Web Docs. (2024).

Php:

1. Se realiza la definición de las variables para la conexión de la base de datos, se usa para establecer la conexión \$conn y un if que permite verificar si la conexión fue exitosa o falló.

2. Se tiene la inclusión del archivo de conexión a la base de datos `include 'db_connect.php';` se establece para la conexión con la base de datos. Encabezados para a la respuesta en

JSON Y CORS mediante header, establece la verificación del método de solicitud en la que se define por una sentencia if, la sanitización y validación de los datos se utiliza filter_var con el filtro FILTER_SANITIZE_STRING para limpiar los datos recibidos, eliminando cualquier carácter especial que no sea válido. Por último, se tiene un cierre de la sentencia y la conexión con \$stmt->close();

\$conn->close();

permite liberar recursos.

Políticas y Privacidad, Términos y Condiciones

Descripción:

Sección que detalla la política de privacidad, y términos y condiciones del sitio.

Tabla 18

Políticas y Privacidad, Términos y Condiciones

Código:	Descripción:
---------	--------------

Htмл:

```

<section class="portada">
  <h1>Privacidad y Política/</h1>
</section>
<main>
  <h1>Se establecen los diferentes artículos, resoluciones y leyes que rigen la aplicación web sacados de la Resoluci
  <h1>El artículo 15 consagra que todas las personas tienen derecho a su intimidad personal, familiar y a su buen nom
  <h1>El artículo 17 de la Ley Estatutaria 1581 de 2012, "Régimen General de Protección de Datos Personales", y el 2º
  <h1>estableció dentro de los deberes de los responsables del tratamiento de datos personales, desarrollar políticas par
  <h1>La Ley 1712 de 2014, sobre transparencia y derecho de acceso a la información pública nacional, adiciona nuevos
  <h1>cual se expide el Decreto Reglamentario Único del Sector Cultura", el cual establece las directrices para la califi
  <h1>y Reservada; (3) Esquema de Publicación de Información; (4) Programa de Gestión Documental.</h1>
  <h1>El artículo 2.2.29.1.2.1. del Decreto 1074 de 2015, subrogado por el artículo 1 del Decreto 767 de 2022, determi
  <h1>seguridad y privacidad de la información en todos sus procesos, trámites, servicios, sistemas de información, infra
  <h1>La Resolución 500 de 2021 del Ministerio de Tecnologías de la Información y las Comunicaciones, establece línea
  <h1>y, establece los lineamientos y estándares para la estrategia de seguridad digital, a los sujetos obligados señalad
  <h1>El artículo 5 de la misma Resolución, establece que los sujetos obligados deben adoptar la estrategia de seguri
  <h1>Privacidad de la información que se integra al Plan de Acción en los términos artículo 2.2.22.3.14. del capítulu
  <h1>de la Información [ ] MSPÍ señalado en el Anexo 1 de la misma resolución, como habilitador de la política de Gobierno
  <h1>El Documento COMPEL 3995 de 2020 formula la Política Nacional de Confianza y Seguridad Digital en la República
  <h1>digital de los ciudadanos, del sector público y del sector privado del país; actualizando el marco de gobernanza en
  <h1>El artículo 22 de la Resolución 0448 de 2022 establece que la política de Seguridad y Privacidad de la Informac
  <h1>así lo requieran, así las cosas y teniendo en cuenta las observaciones y recomendaciones que surgieron del ejercici
  <h1>ARTÍCULO 2. Ámbito de aplicación. La Política de Seguridad y Privacidad de la Información, Seguridad Digital y
  <h1>entidades adscritas y del sector de las Tecnologías de la Información y las Comunicaciones, así como aquellas perso
  <h1>relacionadas que accedan, ya sea interna o externamente a cualquier activo de información, independientemente de su
  <h1>características, a los principios de la Política General de Seguridad y Privacidad de la Información, Seguridad Digital y co
  </h1>

```

Style:

```

myButton {
  .button {
    flex: 1;
    display: flex;
    justify-content: flex-end;
    align-items: center;
    /*Agrega los botones del menú*/
    margin-left: 5px; /* Agrega un margen izquierdo a los botones */
  }
  myButton, myButton1 {
    margin-left: 10px;
  }
  /*Cuerpo de la aplicación parte izquierda */
  main {
    width: 100%; /* Asegura que el main ocupe todo el ancho disponible */
    padding: 60px;
    justify-content: justify;
    align-items: justify;
    text-align: justify;
    overflow-x: hidden; /* Evita la barra de desplazamiento horizontal */
  }
  h1 {
    font-weight: normal; /* Cambia el peso de la fuente a normal */
  }
  /*código de contacto*/
  .contenedor {
    display: flex;
    justify-content: center;
  }

```

Javascript:

```

document.getElementById("myButton").onclick = function () {
  location.href = "Voluntario.html";
};
document.getElementById("myButton1").onclick = function () {
  location.href = "donar.html";
};
const h1Element = document.querySelector('.portada h1');

// Cambia el color de fondo y el tamaño de la fuente
h1Element.style.setProperty('--h1-font-size', '4em');
// Arreglo con los colores de la bandera de Colombia
const coloresBandera = ['#FFD444', '#0033A0', '#CE1126'];
let currentIndex = 0;
// Función para cambiar el color del texto
function cambiarColorTexto() {
  h1Element.style.color = coloresBandera[currentIndex];
  currentIndex = (currentIndex + 1) % coloresBandera.length;
}
// Cambia el color inicialmente y establece un intervalo para cambiar cada 2 segundos
cambiarColorTexto();
setInterval(cambiarColorTexto, 2000); // Cambia cada 2 segundos (2000 milisegundos)

function toggleMenu() {
  const menu = document.querySelector('.mobile-menu');
  menu.classList.toggle('open');
}

```

Htмл: Para el uso del texto se

utilizó el h1, permite que la letra se vea visible, motivando a ser leída.

Style: Este estilo permite

ajustar la letra, el ancho y largo de la pestaña, de modo que los elementos queden alineados dentro de él.

Además, permite que el menú se adapte, añadiendo color y estilo.

Javascript: En esta se

encuentra una redirección al dar clic en botones usando el código:

`document.getElementById("").onclick = function () { location.href = ""; };`

Se evidencia un cambio

dinámico del color del texto en un elemento, en este caso el índice se reinicia al final del arreglo para crear un ciclo continuo, además, se llama la función inicialmente para aplicar el primer color y luego se utiliza `setInterval` para cambiar el color del texto cada 2 segundos (2000

milisegundos). La función para activar/desactivar un menú móvil, Esta función selecciona un elemento con la clase mobile-menu.

“con `menu.classList.toggle('open')`, se añade o remueve la clase open cuando se activa la función. Esto suele usarse para mostrar u ocultar un menú en dispositivos móviles, dependiendo del estilo CSS asociado a la clase open”
Mozilla Developer Network (MDN) Web Docs. (2024).

Encabezado

Descripción:

Este se encontrará en todas las pestañas en la parte inferior.

Tabla 19

Encabezado

Código:	Descripción:
Html:	Html: Como se estructura el código, En primer lugar, se debe poner

```

<footer id="myFooter">
  <div class="column1">
    <div class="redes">
      <h3>¡Síguenos en!</h3>
      <div class="iconos">
        <a href="https://www.facebook.com/profile.php?id=100077443789546" class="sigueface" target="_blank"><i class="fa-brands fa-facebook"></i></a>
        <a href="https://www.instagram.com/trapitosconamor2014?igsh=9b44e0706d0d1a8e" class="sigueinst" target="_blank"><i class="fa-brands fa-instagram"></i></a>
        <a href="https://www.tiktok.com" class="tiktok" target="_blank"><i class="fa-brands fa-tiktok"></i></a>
        <a href="https://www.youtube.com" class="youtube" target="_blank"><i class="fa-brands fa-youtube"></i></a>
      </div>
    </div>
    <div class="cont-footer">
      <a href="#"> Privacidad y Políticas </a> | <a href="#"> Términos y Condiciones </a> |
    </div>
  </div>
  <div class="column2">
    <p>©copy; 2024 | Fundación Trapitos Con Amor | Todos los derechos reservados.</p>
  </div>
</footer>

```

Style:

```

423 .column1 {
424   position: absolute;
425   bottom: 0; /* Coloca el footer al final de la ventana */
426   width: 100%; /* Asegura que el footer ocupe todo el ancho */
427   min-height: 8vh; /* Altura mínima del footer */
428   background: linear-gradient(
429     100deg,
430     rgba(255, 241, 104, 1) 0%,
431     rgba(255, 241, 104, 1) 30%,
432     rgba(25, 108, 236, 1) 40%,
433     rgba(25, 108, 236, 1) 70%,
434     rgba(234, 71, 71) 75%,
435     rgba(234, 71, 71) 100%
436   );
437   color: white; /* Color del texto del footer */
438   text-align: center; /* Alineación del texto en el centro */
439   padding: 10px 0; /* Espaciado interior del footer */
440 }
441
442 .column1 {
443   display: flex;
444   justify-content: space-between;
445   align-items: center;
446   padding: 10px;
447   gap: 100px;
448 }
449
450 .redes {
451   display: flex;
452   align-items: center; /* Centrar verticalmente */
453   gap: 10px; /* Espacio entre el texto y los iconos */
454   width: 100%;
455   padding-left: 40px;
456 }
457 .redes a {
458   margin: 0; /* Eliminar márgenes */
459   transform: translateY(10%); /* Ajustar para centrar verticalmente */
460   color: blue;
461   margin-right: 10px;
462   font-size: 28px;
463 }

```

```

.icons {
  transform: translateY(22%); /*mover el texto o iconos de forma vertical*/
  display: flex;
  gap: 8px; /*se despejan los iconos*/
}

.icons a: hover {
  transform: scale(1.2);
}

.cont-footer {
  width: 60%;
  padding-right: 10px;
  font-size: 20px;
  transform: translateY(10%);
}

.cont-footer a {
  text-decoration: none;
  color: blue;
}

.sigueface, .sigueinst, .tiktok, .youtube {
  color: black;
  font-size: 30px; /* Tamaño del icono */
}

.column2 {
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 10px;
  color: black;
  font-weight: bold;
  font-size: 18px;
  transform: translateX(-15%)
}

```

<footer>. Dentro de este contenedor se colocan cuatro contenedores <div>, cada uno de ellos nombrado con class, el primero permite agrupar la primera columna, que en ella se encontrarán las redes definidas con otro contenedor y dentro de ella se encuentra otro contenedor de iconos, después de ello se evidencian dos botones que mediante otro contenedor se encuentran dos botones definidos por <a>, en la siguiente columna se puede evidenciar un contenedor que dentro de él hay un párrafo.

Style: Este permite definir los colores que elegí para el color de fondo de mi footer; también permite reubicar los botones, el texto y los iconos de una manera visible y organizada, entre otros.

Base de datos

Descripción:

Aquí se explicará cómo fue la construcción de la base de datos mediante códigos y que será manejada por el dueño de la Fundación.

Usuario y Contraseña

Tabla 20

Usuario y Contraseña

Código:	Descripción:
<p>Html:</p> <pre> <!DOCTYPE html> <html lang="es"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Inicio de sesión</title> <link rel="stylesheet" href="style/style.css"> </head> <body> <header> </header> <div class="login-container"> <form id="loginForm" novalidate> <h2>Iniciar sesión</h2> <label for="username">Nombre de usuario:</label> <input type="text" id="username" name="username" required> <label for="password">Contraseña:</label> <input type="password" id="password" name="password" required> <button type="submit">Iniciar sesión</button> <p id="error-message"></p> <!-- Error message container --> </form> </div> <script src="script/script.js"></script> </body> </html> </pre>	<p>Html: En este código se tiene la declaración del documento y lenguaje <code><!DOCTYPE html> <html lang="es"></code>, se encuentra un encabezado del documento <code><head></code>, en el que contiene la codificación de caracteres UTF-8. El cuerpo del documento hay <code><body></code> Inicia el cuerpo de la página, donde se coloca el contenido que verá el usuario,</p> <p><code><header></code> Este elemento contiene dos imágenes que representan logos. Contenedor del formulario de inicio de sesión, crea un contenedor con la base <code>login.container</code> que probablemente se utiliza para dar estilo al formulario de inicio de sesión, <code><label</code></p>
<p>Style:</p>	

```

> style > Styles >
styles.css
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  position: relative;
  background-color: #f0f0f0;
  background-image: url(../img/logo.png); /* Ruta de la imagen de fondo */
  background-size: cover; /* Asegura que la imagen cubra toda la pantalla */
  background-position: center; /* Centra la imagen en la pantalla */
  background-repeat: no-repeat; /* Evita que la imagen se repita */
  background-attachment: fixed; /* Fija la imagen en su lugar al hacer scroll */
}

header {
  position: absolute;
  top: 0;
  width: 100%;
  display: flex;
  justify-content: space-between;
  padding: 10px;
  box-sizing: border-box;
}

.logo {
  height: auto;
}

.logoIzquierda {
  margin-top: -60px;
  width: 300px; /* Ajusta el tamaño según tus necesidades */
  height: 200px;
}

.logoDerecha {
  width: 200px; /* Ajusta el tamaño según tus necesidades */
}

.login-container {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

form {
  background-color: #fff;
  padding: 40px; /* Ajusta el padding según tus necesidades */
  border-radius: 8px;
  box-shadow: 0 0 10px #ccc;
  width: 500px; /* Ajusta el ancho según tus necesidades */
  box-sizing: border-box; /* Asegura que el padding se incluya en el tamaño total */
  display: flex;
  flex-direction: column; /* Asegura que los elementos se alineen verticalmente */
  align-items: stretch; /* Asegura que los elementos hijos llenen el ancho del formulario */
}

h2 {
  font-size: 2.6em;
  margin: 0 0 20px 0;
  width: 100%; /* Ajusta el ancho del formulario */
  text-align: center; /* Centra el texto del encabezado */
}

label {
  font-size: 1.4em;
  display: block; /* Asegura que las etiquetas estén en una línea nueva */
  margin: 10px 0 5px 0;
  width: 100%; /* Ajusta el ancho del formulario */
}

```

for="username">Nombre de usuario:</label> y
<label for="password">Contraseña:</label>
etiquetas asociadas a los campos de entrada para
el nombre de usuario y contraseña, <button
type="submit">Iniciar sesión</button> Botón
para enviar el formulario.

Style: Este permite definir los colores
que elegí para el color de fondo de mi base de
datos; también permite reubicar los botones, el
texto y los iconos de una manera visible y
organizada, entre otros.

Javascript:

```

document.addEventListener('DOMContentLoaded', function () {
  const loginForm = document.getElementById('loginForm');
  const errorMessage = document.getElementById('error-message');

  loginForm.addEventListener('submit', function (event) {
    event.preventDefault(); // Prevenir el envío del formulario por defecto

    const username = document.getElementById('username').value.trim();
    const password = document.getElementById('password').value.trim();

    // Enviar la solicitud al servidor
    fetch('http://localhost:3000/login', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        'username': username,
        'password': password
      })
    })
    .then(response => response.json())
    .then(data => {
      if (data.success) {
        window.location.href = 'dashboard.html'; // Redirigir a otra página
      } else {
        errorMessage.textContent = data.message;
        errorMessage.style.color = 'red';
      }
    })
    .catch(error => {
      errorMessage.textContent = 'Error al realizar la solicitud.';
      errorMessage.style.color = 'red';
    });
  });
});

```

Javascript:

document.addEventListener('DOMContentLoaded', function () { ... })), se tiene un evento que garantiza que el código se ejecuta solo después de que todo el contenido de la página (HTML) se haya cargado completamente. Se tiene un código para agregar un manejador de eventos para cuando se envía el formulario.
event.preventDefault(); previene el comportamiento predeterminado del formulario,

que es enviar los datos y recargar la página.

Captura los datos ingresados, los envía al servidor usando fetch.

Tablas de la información registrada

Descripción:

En este se explicará de cómo está programado las tablas y el menú.

Tabla 21

Tablas de la información registrada

Código:

Html:

```

<!-- Tabla de Voluntarios Altruistas: />
<!-- Tabla de datos -->
<table id="voluntarios-table">
  <thead>
    <tr>
      <th>ID/>
      <th>Username/>
      <th>Teléfono/>
      <th>Departamento/>
      <th>Municipio/>
      <th>Días/Meses/>
      <th>Actividad/>
      <th>Fecha último/>
    </tr>
  </thead>
  <tbody>
    <!-- Los datos se insertarán aquí mediante PHP -->
  </tbody>
</table>
<!-- Contenedor para los formularios de insertar, actualizar y borrar -->
<div class="form-container">
  <!-- Formulario para insertar datos (izquierda) -->
  <div class="insert-form">
    <h2>Insertar Nuevo Voluntario />
  </div>

```

Style:

Descripción:

Html: Se tiene un encabezado con

<header> contiene un botón que ejecuta una función JavaScript toggleMenu (), esta permite ver un menú de navegación que contiene enlaces a distintas tablas en la base de datos, como voluntarios altruistas, extranjeros, de otra entidad, entre otros, así como el enlace para cerrar sesión. En la sección principal con código <main> que esta permite ver los campos correspondientes a la tabla en la que los datos están registrados dinámicamente con node.js, también, se encuentran tres

```

body {
  width: 1333px;
  margin: 0;
  font-family: Arial, sans-serif;
}
header {
  background: linear-gradient(90deg, rgba(255, 241, 104) 10%, rgba(255, 0, 0, 0.7) 60%);
  padding: 30px;
}
.header-content {
  display: flex;
  align-items: center; /* Centra verticalmente */
  justify-content: space-between; /* Distribuye el espacio entre los elementos */
  padding: 10px;
}
.logo {
  display: flex;
  align-items: center; /* Centra verticalmente */
  justify-content: center; /* Centra horizontalmente */
}
.logo img {
  width: 88px; /* Ajusta el tamaño según lo necesites */
  height: auto; /* Mantiene la proporción */
  margin-left: 100px; /* Espacio entre la imagen y el texto */
}
.logo span {
  color: white;
  margin-left: 200px;
  font-size: 3.2em; /* Ajusta el tamaño del texto */
  text-shadow: 2px 2px 2px rgba(0, 56, 168, 0.7); /* Sombra azul */
}
.menu-toggle {
  margin-left: 50px;
  display: block; /* Asegúrate de que el botón se muestre */
  background: none;
  border: none;
}

```

Javascript:

```

function toggleMenu() {
  const menu = document.getElementById('side-menu');
  menu.classList.toggle('open');
}

document.addEventListener('DOMContentLoaded', function() {
  actualizarTabla();
});

function actualizarTabla() {
  fetch('http://localhost:3000/api/voluntarios')
    .then(response => response.json())
    .then(data => {
      if (!Array.isArray(data) || data.length === 0) {
        console.log('No data available');
        return;
      }
      const tableBody = document.getElementById('voluntarios-table').getElementsByTagName('tbody')[0];
      tableBody.innerHTML = '';
      data.forEach(voluntario => {
        const row = document.createElement('tr');
        row.innerHTML = `
          <td>${voluntario.id}</td>
          <td>${voluntario.username}</td>
          <td>${voluntario.telefono}</td>
          <td>${voluntario.departamento}</td>
          <td>${voluntario.municipio}</td>
          <td>${voluntario.dias_meses}</td>
          <td>${voluntario.motivo}</td>
          <td>${voluntario.actividad}</td>
          <td>${voluntario.fecha_ultimo}</td>
        `;
        tableBody.appendChild(row);
      });
    })
    .catch(error => console.error('Error al obtener la lista de voluntarios:', error));
}

// Actualizar la tabla después de insertar, actualizar o borrar
function refreshTable() {
  actualizarTabla();
}

document.getElementById('insert-form').addEventListener('submit', function(event) {
  event.preventDefault();
  const formData = new FormData(event.target);
  const jsonData = {};
  formData.forEach((value, key) => jsonData[key] = value);

  fetch('http://localhost:3000/api/voluntarios', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(jsonData)
  })
    .then(response => response.json())
    .then(data => {

```

formularios insertar, actualizar y eliminar

se establece por las siguientes secciones:

<form>, <label>, <input>, <option>,

<textarea>, <select> y <button>.

Style: Este permite definir los

colores que elegí para el color de fondo de

mi base de datos; también permite reubicar

los botones, el texto y los iconos de una

manera visible y organizada, entre otros.

Javascript: Este código lo maneja

un sistema de CRUD (crear, leer,

Actualizar, Borrar). Se tiene una función

toggleMenu, esta función se usa para abrir

o cerrar un menú lateral (side-menu), usa

classList.toggle() para añadir o quitar la

clase open del elemento con ID side-menu.

DOMContentLoaded, cuando la página

termina de cargarse, se llama a la función

actualizarTabla (), que obtiene y muestra

los datos de los voluntarios. Se tiene una

refreshTable(), realiza un llamado

nuevamente a actualizarTabla() lo que hace es refrescar la tabla después de una operación sea insertar, actualizar o borrar. El formulario de insertar se maneja usando submit evento listener, sirve para detectar cuando el formulario es enviado que lo hace el siguiente código: document.getElementById('insert-form').addEventListener('submit', function(event) {.

El formulario actualizar es algo similar a inserción solo que en este caso se usa PUT. En el formulario de eliminación, Este formulario permite eliminar un voluntario usando su ID. Se hace una petición DELETE a la API.

Servidor

Descripción:

El código para mandar datos de la base de datos a la tabla

Tabla 22

Servidor

| Código: | Descripción: |
|---|---|
| <pre> // Función para conectar a la base de datos const db = mysql.createConnection({ host: 'localhost', user: 'root', password: '', database: 'usuariosdb' }); console.log('Conexión exitosa a la base de datos. '); return db; } catch (err) { console.error('Error de conexión a la base de datos: \$err.message'); console.error('Código de error: \$err.code'); console.error('Número de error: \$err.errno'); process.exit(1); } // Función para manejar el inicio de sesión async function login(req, res) { const db = await conectarDB(); const { username, password } = req.body; console.log('Recibido username: \$username y contraseña: \$password'); const query = 'SELECT password FROM usuarios WHERE username =?'; const [results] = await db.promise().query(query, [username]); if (results.length === 0) { res.json({ success: false, message: 'Nombre de usuario no encontrado' }); return; } const hashedPassword = results[0].password; console.log('Contraseña hashada a la consulta: \$hashedPassword'); </pre> | <p>Javascript: Se usa un código node.js con xampp, en el que en la aplicación visual studio mediante una terminal se escribe el código node (nombre del archivo.js), permitiendo dar el localhost para ingresar a la base de datos. Se tiene dependencias y configuración básica, se importan las bibliotecas necesarias express (para manejar el servidor web), body-parser (para procesar datos JSON y formularios), mysql2/promise (para conectar y ejecutar consultas a la base de datos MySQL), y bcrypt (para manejar la encriptación de contraseñas), esta configura con el fin de escuchar el puerto 3000 procesando las solicitudes JSON y servir archivos estáticos desde la carpeta Public, para conectar la base de datos se usa la función conectarDB. En el manejo de login permite a los usuarios autenticarse, primero se verifica si el nombre de usuario existe en la base de datos, si existe, se obtiene la contraseña</p> |

almacenada (que está encriptada con bcrypt) y se compara con la contraseña ingresada por el usuario usando bcrypt.compare, dependiendo el resultado con un mensaje JSON indicando éxito o error.

Rutas para los voluntarios

(CRUD):

1. Obtener todos los voluntarios (GET /api/voluntarios): Se ejecuta una consulta para obtener todos los registros de la tabla voluntarios altruista y se devuelve como respuesta en formato JSON.

2. Insertar un nuevo voluntario (POST /api/voluntarios): Valida si todos los campos obligatorios están presentes. Luego, inserta el nuevo registro en la base de datos y reasigna los IDs para mantenerlos secuenciales. Finalmente, devuelve el nuevo voluntario insertado.

3. Actualizar un voluntario (PUT /api/voluntarios/:id): Valida los campos obligatorios, actualiza

los datos del voluntario basado en su id y devuelve un mensaje de éxito.

4. Borrar un voluntario (DELETE /api/voluntarios/:id): Borra un voluntario por su id y reasigna los IDs secuenciales. Devuelve los datos actualizados de la tabla.

5. Manejo de voluntarios extranjeros: Similar a la tabla voluntarios altruista, pero con la tabla voluntarios extranjeros, manejando inserciones, actualizaciones, y eliminación de registros. También se reasignan los IDs secuenciales después de cada operación.

6. Manejo de voluntarios de otra entidad: Se maneja una tabla adicional llamada

7. voluntarios_otra_entidad, con operaciones CRUD similares a las tablas anteriores, pero con campos más específicos relacionados con los voluntarios de otra entidad (por ejemplo, información de la entidad y contacto).

Entre otras que se tiene aquí en el código de la base de datos.

Reasignación de IDs

secuenciales: Cada vez que se inserta o borra un voluntario, el código reasigna los IDs de manera secuencial para mantener el orden correcto de los registros. Este código gestiona múltiples tablas y maneja operaciones completas de creación, lectura, actualización y eliminación (CRUD), asegurando además la seguridad de las contraseñas de los usuarios mediante encriptación.

Express es un framework de Node.js que facilita la creación de aplicaciones web y APIs RESTful. La documentación oficial proporciona guías y ejemplos sobre cómo definir rutas, manejar peticiones, y gestionar middleware.

<https://expressjs.com/>

CRUD (Create, Read, Update, Delete).

https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

Middleware usado para analizar los datos de las peticiones entrantes en el formato JSON o URL-encoded. <https://www.npmjs.com/package/body-parser>

Explicación código de ingresar contraseña nueva y usuario

Tabla 23

Ingresar contraseña nueva y usuario

Código:	Descripción:
<p>Javascript</p> <pre> const bcrypt = require('bcrypt'); const mysql = require('mysql/promise'); async function registrarUsuario() { const username = 'karen12'; const password = '150916'; const db = await mysql.createConnection({ host: '127.0.0.1', user: 'root', password: '', database: 'voluntariodb' }); bcrypt.genSalt(10, (err, salt) => { if (err) { console.error(err); } else { bcrypt.hash(password, salt, (err, hash) => { if (err) { console.error(err); } else { // Almacenar el usuario y la contraseña hashada en la base de datos db.execute("INSERT INTO usuarios (username, password) VALUES (?, ?)", [username, hash]) .then((results) => { console.log('Usuario registrado con éxito'); }) .catch((err) => { console.error(err); }) .finally(() => { db.end(); }); } }); } }); } registrarUsuario(); </pre>	<p>Javascript: Almacena un usuario y contraseña de forma segura, se usa para el hash de la contraseña bcrypt. La importación de módulos se tiene como principal bcrypt que es conocida por su seguridad, ya que aplica un algoritmo de hash lento. Se utiliza una función asincrónica registrarUsuario () esta permite manejar el proceso de registrar un usuario. Como definición de usuario y contraseñas se utiliza const para indicar cuál contraseña y usuario quiero que quede. Se tiene la generación de sal para el hash, la sal es un</p>

valor aleatorio añadido a la contraseña antes de hashearla para proteger contra ataques de diccionario. Inserción del usuario y contraseña hasheada en la base de datos, se ejecuta una consulta SQL para insertar el nombre de usuario y la contraseña hasheada en la tabla usuarios de la base de datos. ¿El uso de? como marcadores de posición en la consulta ayuda a evitar inyecciones SQL, ya que los valores de username y hash se pasan de manera segura. Para el llamado a la función, registrarUsuario (), se usa para que todo el proceso de hash de contraseña y almacenamiento en la base de datos se ejecute.

Software (Gestión de datos)

Descripción:

Aquí se podrá visualizar el código usado para realizar el software sencillo para la fundación.

Conexión de base de datos

Descripción:

Se mostrará el código que permitirá conectar mi interfaz con la base de datos.

Tabla 24

Conexión de base de datos

Código:	Descripción:
<pre> 1 /** 2 package swbdfund; 3 /** 4 * 5 * @author Karen Hortua 6 */ 7 import java.sql.Connection; 8 import java.sql.DriverManager; 9 import java.sql.SQLException; 10 11 public class ConexionBD { 12 static String basededatos = "beneficiarios"; 13 static String usuario = "root"; // El nombre de usuario por defecto es 14 static String password = ""; // Por defecto, XAMPP no tiene contraseña 15 static String url = "jdbc:mysql://localhost/"+basededatos; 16 static Connection conexionBD = null; 17 18 public ConexionBD() {} 19 20 public static Connection getConnection() { 21 if (conexionBD == null) { 22 try { 23 conexionBD = DriverManager.getConnection(url, usuario, 24 } catch (SQLException e) { 25 e.printStackTrace(); 26 } 27 } 28 return conexionBD; 29 } 30 } 31 } </pre>	<p>Java: En el código de paquete y autor, se usa con paquete swbdfund, y en la parte de abajo marca como Karen Hortua la autora de la clase. Se importan las clases necesarias para manejar conexiones SQL:</p>
	<ol style="list-style-type: none"> 1. Connection:
	<p>Representa la conexión a la base de datos.</p>
	<ol style="list-style-type: none"> 2. DriverManager:
	<p>Administra las conexiones con las bases de datos.</p>
	<ol style="list-style-type: none"> 3. SQLException:
	<p>Maneja errores relacionados con SQL.</p>
	<p>Declaración de la clase</p>
	<p>conexiónBD, se encarga de gestionar la conexión</p>

a la base de datos. Se tiene unas variables estáticas, que son las que declaran las variables que contiene la información necesaria para conectar. Se tiene un método `getConexion`, se usa para retornar la conexión a la base de datos, si la conexión aún no existe (`conexionBD == null`), la crea usando `DriverManager.getConnection`. Método `desconectar`, cierra la conexión si está abierta. Llama a `conexionBD.close()` para cerrar la conexión y la establece nuevamente como `null` para evitar reusos. Si hay un error, lo captura y lo imprime.

XAMPP Official Documentation.

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

Oracle Documentation - JDBC API.

<https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/>

MySQL Connector/J Documentation

<https://dev.mysql.com/doc/connector-j/8.0/en/>

Tabla 25

Login Software

Código:	Descripción:
<pre> package svbdfund; import javax.swing.*; import java.awt.*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.*; public class LoginPanel extends JFrame implements ActionListener { private JTextField tfUsername; private JPasswordField pfPassword; private JButton btLogin; public LoginPanel() { setTitle(title: "Login"); setSize(width: 400, height: 200); setDefaultCloseOperation(operation: EXIT_ON_CLOSE); setLocationRelativeTo(c: null); setLayout(new GridBagLayout()); GridBagConstraints gbc = new GridBagConstraints(); gbc.insets = new Insets(top: 10, left: 10, bottom: 10, right: 10); gbc.fill = GridBagConstraints.HORIZONTAL; JLabel lUsername = new JLabel(text: "Username:"); gbc.gridx = 0; gbc.gridy = 0; add(comp: lUsername, constraints: gbc); tfUsername = new JTextField(columns: 20); gbc.gridx = 1; pfPassword = new JPasswordField(columns: 20); gbc.gridx = 1; gbc.gridy = 1; add(comp: pfPassword, constraints: gbc); btLogin = new JButton(text: "Login"); gbc.gridx = 1; gbc.gridy = 2; gbc.anchor = GridBagConstraints.EAST; add(comp: btLogin, constraints: gbc); btLogin.addActionListener(l: this); setVisible(b: true); } @Override public void actionPerformed(ActionEvent e) { if (e.getSource() == btLogin) { String username = tfUsername.getText(); String password = new String(pfPassword.getPassword()); if (authenticate(username, password)) { new Tablabeneficiarios(); dispose(); } else { JOptionPane.showMessageDialog(parentComponent: this, message: "In } } } } </pre>	<p>Java: Este código sirve para interfaz gráfica de usuario (GUI) para un sistema de login utilizando Swing y se conecta a una base de datos. Se tiene una clase loginPanel, permite crear una ventana de login y responder a eventos como dar clic en el botón. Componentes GUI, este permite crear etiquetas JLabel, campos de texto JTextField para nombre de usuario y JPasswordField para la contraseña, y un botón JButton, los componentes se añaden a la ventana usando GridBagLayout. Se tiene un método actionPerformed que permite cuando el usuario hace clic en el botón de login registra en la base de datos, el nombre de usuario y la contraseña ingresados. El método authenticate para verificar si los datos son correctos, Prepara una consulta SQL para verificar si existe un</p>

usuario con el nombre de usuario y la contraseña proporcionados. El método main, inicia la aplicación llamando a LoginPanel en un hilo de interfaz gráfica con SwingUtilities.invokeLater.

Horstmann, C. S., & Cornell, G. (2019). Core Java Volume I—Fundamentals (11th ed.). Prentice Hall.

<https://www.vitalsource.com/products/core-java-volume-i-fundamentals-cay-s-horstmann-v9780135167182>

Interfaz de la tabla

Tabla 26

Interfaz de la tabla

Código:

Descripción:

Es una interfaz gráfica para

gestionar los beneficiarios en una base de datos. Se tiene unas librerías, que permiten crear componentes gráficos como JTable, JLabel, JButton, etcétera, JDBC maneja la conexión con una base de datos MYSQL. Se tiene una clase Tablabeneficiarios, dentro ella hay JFrame que se usa una ventana con una tabla y un formulario. Se utiliza unos componentes gráficos que incluyen etiquetas JLabel, campos de texto JTextField, y botones JButton para la gestión de los datos de beneficiarios.

Diseño de interfaz, se crean paneles y se configuran diferentes posiciones y tamaños para los elementos (botones, textos, etcétera). Conexión a la base de datos, se usa DriverManager para conectar a la base de datos, el método mostrarDatos() consulta los datos de los beneficiarios y los muestra en la tabla. Interacción del usuario, se implementa la interfaz ActionListener para manejar eventos, de modo que cuando

```
@Override
public boolean isCellEditable(int row, int column) {
    return false; // Todas las celdas no editables
}
}
table = new JTable(mo_model);
JScrollPane scroll = new JScrollPane(table);
scroll.setBounds(15, 15, width: 850, height: 280);
panel.add(scroll);

// Modelo de la tabla, especificando que las celdas no son editables
DefaultTableModel model = new DefaultTableModel(columnNames, rowCount) {
    @Override
    public boolean isCellEditable(int row, int column) {
        return false; // Todas las celdas no son editables
    }
};

// Centrar el texto en las celdas
DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();
centerRenderer.setHorizontalAlignment(JLabel.CENTER);
for (int i = 0; i < table.getColumnCount(); i++) {
    table.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
}

// Configuración de los elementos del formulario
JLabel tituloInsertar = new JLabel("INSERTAR DATO");
tituloInsertar.setBounds(50, 300, width: 100, height: 20); // Separado del lado izquierdo
panel.add(tituloInsertar);

JLabel tituloActualizar = new JLabel("ACTUALIZAR DATO");
tituloActualizar.setBounds(355, 300, width: 200, height: 20);
```

el usuario presiona un botón.

Personalización de la tabla, los datos se centran en las celdas usando un `DefaultTableCellRenderer`.

Código para uso de dispositivo móvil

Este código permitirá tener un diseño del tamaño de una Tablet o celular.

Tabla 27

Uso de dispositivo móvil

Código:	Descripción:
<p>Html:</p> <pre> <!-- Icono de menú para dispositivos móviles --> <div class="menu-icon" onclick="toggleMenu()"> <div class="bar1"></div> <div class="bar2"></div> <div class="bar3"></div> </div> <!-- Menú oculto para dispositivos móviles --> <nav class="mobile-menu"> Inicio <li class="dropdown"> Ayuda <svg class="arrow" xmlns="http://www.w3.org/2000/svg" width= <div class="dropdown-content"> ¿Quiénes Somos? Preguntas Frecuentes </div> Testimonios ¿Qué hacemos? Voluntario Donar </nav> </pre>	<p>Html: Se tiene el icono de menú para dispositivos móviles, este bloque de código representa el icono de hamburguesa que es común en las versiones móviles <code><div class="menu-icon" onclick="toggleMenu()"></code>, <code>bar1</code>, <code>.bar2</code>, <code>.bar3</code> son los tres elementos visuales que forman las líneas horizontales del icono de hamburguesa. La función <code>toggleMenu()</code> “es un evento que se activa al hacer clic en el ícono y se encarga de mostrar u ocultar el menú. Esta función probablemente cambia las clases o el estilo del menú para</p>
<p>Javascript:</p> <pre> function toggleMenu() { const menu = document.querySelector('.mobile-menu'); menu.classList.toggle('open'); } </pre>	
<p>Style:</p>	

```

/* Media queries para dispositivos más pequeños */
@media only screen and (max-width: 538px) {
  .menu {
    display: none; /* Oculta el menú principal en dispositivos móviles */
  }
  button {
    display: none;
  }
  .mobile-menu {
    text-align: center;
    display: none; /* Oculta inicialmente el menú móvil */
    position: absolute;
    top: 100px; /* Ajusta la posición superior del menú móvil */
    right: 30px;
    width: 33%;
    background-color: #000000; /* Color de fondo del menú móvil */
    z-index: 1000; /* Z-index para estar encima del contenido */
    padding: 10px;
    box-sizing: border-box; /* Incluye el padding en el ancho total */
  }
  .mobile-menu.open {
    display: block; /* Muestra el menú móvil cuando está abierto */
  }
  .mobile-menu ul {
    list-style: none;
    padding: 0;
    margin: 0;
  }
}

```

hacerlo visible o invisible”- Mozilla Developer Network (MDN). (s.f.)

<https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>

ner . Este menú oculto para dispositivos

móviles, : Crea una lista de enlaces

(elementos), que representan las opciones

del menú. Las clases como "dropdown" y

"dropdown-content" están diseñadas para

manejar la visualización del submenú, y

probablemente se muestran/ocultan usando

estilos CSS o JavaScript.

JavaScript: `Document.querySelector()`, esta función selecciona el primer elemento del documento que coincide con el selector CSS que se le pasa como argumento.

En este caso, selecciona el elemento con la clase `mobile-menu`, que es el menú de navegación móvil. El resultado se almacena en el constante `menú`. `classList.toggle()`: Este método añade o elimina una clase en un elemento dependiendo de si ya existe o no.

Style: Permite desde este código

@media only screen and (max-width: 538px)

realizar el ajuste de ancho y largo de la

pantalla, esta función me permite ajustar los

diferentes diseños y colores para que se

muestre en un dispositivo móvil y Tablet.

Requisitos no Funcionales

Rendimiento

Descripción:

Tiempo de respuesta de las páginas inferior a 3 segundos.

Seguridad

Descripción:

Implementar HTTPS, CRUD, la sal, bcrypt.

Usabilidad

Descripción:

Diseño adaptable para dispositivos móviles.

Estabilidad

Descripción:

Estructura modular que permita añadir nuevas funcionalidades en el futuro.

Compatibilidad

Descripción:

Soporte para los navegadores web más populares (Chrome, Firefox, Safari, Edge) y programas para descargar aplicaciones.

Diseño

Nota: En este punto se establecerá una breve descripción e imágenes del diseño de las pestañas del aplicativo web.

Diseño de aplicativo web

Página Principal

Descripción:

(Parte 1) Aquí se verá en la parte superior los botones que dirigen a diferentes pestañas, con una segunda portada que muestra el lema "Ayúdanos a ayudar" en la parte izquierda, representando los tres colores de la bandera colombiana, en la derecha una imagen de los fundadores de la fundación. También se muestra otra portada que de manera transitoria exhibe diferentes frases con su respectivo autor. **(Parte 2)** En el main de la pestaña izquierda se destaca un carrusel que tiene como objetivo ofrecer información sobre la fundación. **(Parte 4)** Debajo de este, se encuentra una pequeña descripción de los detalles de la fundación, con una opción para mostrar más información. **(Parte 5)** Más abajo, hay otro carrusel que muestra los diferentes extranjeros que visitan la fundación, y finalmente, un formulario de comentarios que permite comentar de manera anónima o con nombre, además de impedir el uso de malas palabras. **(Parte 3)** En el main de la parte derecha se ubican los diferentes iconos de las redes sociales y correo, para contactarse con la fundación. En la parte central se presentan videos sobre la fundación y la ubicación del pueblo en donde se encuentra, y, por último, un botón que permite redirigir a WhatsApp para chatear, interactuar y resolver inquietudes sobre la fundación. **(Parte 6)** En el footer, se destacan los tres colores de la bandera colombiana, las redes sociales para que los sigan, y los enlaces a la privacidad, políticas, términos y condiciones.

Tabla 28

Página Principal diseño

Parte 1



Parte 2



Parte 3

Contacto:



Parte 4



DETALLES DE LA FUNDACIÓN


Recibimos y entregamos donaciones de todo tipo nuevo y usado, Ejemplo: ropa, zapatos, juguetes, muebles, uniformes, accesorios en general, comida, dinero y mucho más....Ofrecemos recreación, clases de idiomas, celebramos fechas especiales...

[¿Qué hacemos?](#)




Parte 5

VOLUNTARIOS EXTRANJEROS



RECIBIMOS Y ENTREGAMOS TODO TIPO DE DONACIONES. LOS INTERESADOS EN SER PARTE DE ESTA CAUSA, SIN PÁGINO DE LUCRO, BIENVENIDOS. YA SEAN DONANTES O BENEFICIARIOS RECIBIMOS ROPA, ZAPATOS, INFORMATICA, ARTICULOS PARA NIÑOS Y ADULTOS, TUVIETES Y DEMAS COSAS QUE PUEDEN UTILIZARSE.



Deja Tú Comentario


Selecciona el tipo de comentario:

Anónimo ▼

Comentario:

[Enviar Comentario](#)

Visítanos:



Parte 6

Síguenos en: [f](#) [@](#) [🎵](#) [📺](#)

[Privacidad y Políticas](#) | [Términos y Condiciones](#) |

© 2024 | Fundación Trapitos Con Amor | Todos los derechos reservados

Ayuda

¿Quiénes Somos?

Descripción:

(Parte 1) En la parte superior se encuentran los botones que dirigen a diferentes pestañas. A continuación, se muestra una segunda portada que destaca el título y una imagen con los fundadores. **(Parte 2)** En el main, en la parte derecha, hay un menú desplegable que permite acceder a la historia, misión, visión, fundadores, colaboradores y pensamientos de la fundación. **(Parte 3)** mostrando la información correspondiente en la parte izquierda del main. **(Parte 2)** Además, dentro del menú desplegable hay una frase con su autor. **(Parte 4)** En el footer, se destacan los tres colores de la bandera colombiana, las redes sociales para que los sigan, y los enlaces a la privacidad, políticas, términos y condiciones.

Tabla 29

Quienes Somos diseño

Parte 1



Parte 2





Parte 3



Historia

Es una fundación sin ánimo de lucro, cuyo lema es... Ayúdanos a Ayudar. Nace aproximadamente en el año 2017 con el nacimiento de nuestro hijo (quien fue el impulsor de esta causa) con su natalicio llegaron las primeras donaciones las cuales nos abrieron un gran interrogante.... "¿Por qué botar las cosas que otras personas pueden utilizar? Hoy en día, de base sin contar con las foráneas, son más de 350 familias beneficiarias, entre las que cuentan toda clase de grupos, como son: población rural, Campesina, Madres adolescentes (muchas de ellas con varios bebés a su cargo), desplazados de la violencia en Colombia, víctimas del conflicto armado en Colombia, Inmigrantes (mayormente del país vecino "Venezuela"), Madres cabeza de familia, personas de avanzada edad en estado de Alta vulnerabilidad, discapacitados corporal y psicológico, en fin toda clase de personas sin distinción alguna de género, etnicidad, orientación sexual, religiosa, o política. Estamos ubicados en un municipio de Colombia llamado CÁQUEZA, al oriente de Cundinamarca (en medio de Bogotá y Villavicencio). En su mayoría su extensión es de población rural y de escasos recursos económicos.

Parte 4

síguenos en:    

| Privacidad y Políticas | Términos y Condiciones |

© 2024 | Fundación Trapitos Con Amor | Todos los derechos reservados

Preguntas Frecuentes

Descripción:

(Parte 1) En la parte superior se encuentran los botones que dirigen a las diferentes pestañas. A continuación, se presenta una segunda portada que destaca el título, el cual cambia dinámicamente los colores de la bandera colombiana, sobre un fondo con la imagen de los fundadores. **(Parte 2)** En la parte derecha del main, hay un apartado de preguntas con una flecha que, al desplegarse, revela las respuestas. **(Parte 3)** Además, en esta misma sección de la parte izquierda, se ubican los íconos de las redes sociales y del correo electrónico para contactar a la fundación. Debajo de estos, se muestran videos relacionados con la fundación. **(Parte 4)** En el footer, se destacan los tres colores de la bandera colombiana, las redes sociales para que los sigan, y los enlaces a la privacidad, políticas, términos y condiciones.

Tabla 30

Preguntas Frecuentes diseño

Parte 1:



Parte 2:

Parte 3:

¿Cómo puede ser Voluntario en la Fundación?

▼

Hace clic en el botón de voluntario, selecciona uno de los formularios que cumpla con las expectativas que busca en la fundación. Si se encuentra cerca de la fundación o en un pueblo cercano, también podrá realizar el trámite en la sede.

¿A qué personas ayuda la Fundación?

▼

Ayuda a población rural, Campesina, Madres adolescentes (muchas de ellas con varios bebés a su cargo), desplazados de la violencia en Colombia, víctimas del conflicto armado en Colombia, Inmigrantes (mayormente del país vecino "Venezuela"), Madres cabeza de familia, personas de avanzada edad en estado de Alta vulnerabilidad, discapacitados corporal y psicológico, en fin toda clase de personas sin distinción alguna de género, etnicidad, orientación sexual, religiosa, o política.

¿Cómo donar en la fundación?

▼




¿Cuál es el impacto del apoyo?


▼


¿Cómo se puede recibir un mercado del banco de alimentos


▼

Contacto:







Parte 4:

siguenos en:    

| Privacidad y Políticas | Terminos y Condiciones |

© 2024 | Fundación Trapitos Con Amor | Todos los derechos reservados

Testimonios

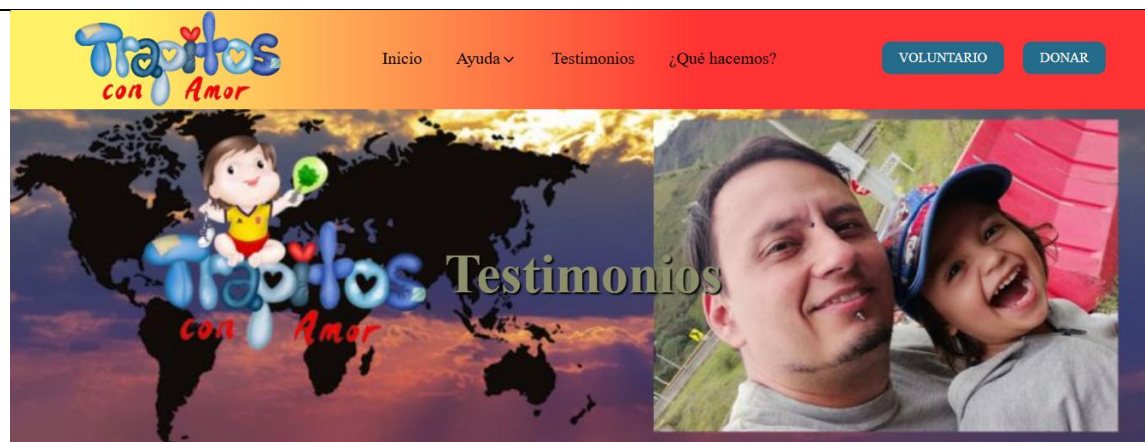
Descripción:

(Parte 1) En la parte superior se encuentran los botones que dirigen a las diferentes pestañas. A continuación, se presenta una segunda portada que destaca el título, el cual cambia dinámicamente los colores de la bandera colombiana, sobre un fondo con la imagen de los fundadores. **(Parte 2)** En la sección del main se encuentra diferentes cuadros con una imagen en la resalta el agradecimiento y cariño que le tienen a la fundación. **(Parte 3)** En el footer, se destacan los tres colores de la bandera colombiana, las redes sociales para que los sigan, y los enlaces a la privacidad, políticas, términos y condiciones.

Tabla 31

Testimonios diseño

Parte 1:



Parte 2:

<p>Realmente es grato saber que en estas dificultades contamos con el valioso apoyo de una persona que con gran empeño realiza el duro trabajo de recolectar todo lo que para algunos quizás es poco pero para otros como yo es mucho, gracias a su gran labor y a su noble corazón mi familia y yo recibimos esa ayuda de parte de trapitos con amor, gracias señora luz marina por compartir con nosotros sus logros, gracias y Dios bendiga a todas las personas que hacen sonreír una familia con los mercados y demás cosas que dan a cada uno de nosotros, Dios la proteja y bendiga su familia!</p> <p>12:06 a. m</p>		
<p>Gladys Salazar con Cedula numero la presente hago constar que tengo conocimiento de la fundación trapitos con amor que he hecho de mucha ayuda para</p> <p>harmosa labor estoy agradecida</p>	<p>Me siento muy agradecida y bendecida por esta bella labor que hace esta grandiosa fundacion TRAPITOS CON AMOR gracias a ellos e tenido ropa, comida y juguetes para mi hija y porsupuesto clases</p>	<p>Escribo de ante mano para agradecer a la fundacion trapitos con amor ya que gracias a la labor de apoyo que brindan ala comunidad me he visto beneficiada al igual que mis tres hijos su ayuda sin ningun interes lucrativo siempre estan dispuestos</p>

Parte 3:

siguenos en:

© 2024 | Fundación Trapitos Con Amor | Todos los derechos reservados

[Privacidad y Políticas](#) | [Terminos y Condiciones](#)

¿Qué hacemos?

Descripción:

(parte 1) En la parte superior se encuentran los botones que dirigen a las diferentes pestañas. A continuación, se presenta una segunda portada que destaca el título, el cual cambia

dinámicamente los colores de la bandera colombiana, sobre un fondo con la imagen de los fundadores. **(parte 2)** En la sección de main, se encuentran subtítulos que identifican las diferentes temáticas de la fundación, acompañados de imágenes que ilustran las actividades realizadas en cada una de ellas. **(parte 3)** En el footer, se destacan los tres colores de la bandera colombiana, las redes sociales para que los sigan, y los enlaces a la privacidad, políticas, términos y condiciones.

Tabla 32

¿Qué hacemos? diseño

Parte 1:



Parte 2:

- Recibo y Entrega de todo tipo de donaciones.



- Entrega de mercados del banco de alimentos.



- Intercambio Cultural.





Invitamos amigos de todas partes del mundo a que nos ayuden a interactuar con los beneficiarios de la fundación. (Los gastos de manutención por el tiempo compartido son a cuenta nuestra "alimentación, servicios públicos"). Clases de Inglés para las personas de la fundación que posiblemente nunca en su vida tendrán contacto directo con un extranjero. Lo cual hace que cambie en algo su perspectiva de vida hacia un futuro y posiblemente vean la vida de otra forma.

* Clases de Inglés

Las clases van dirigidas y son recibidas para todo tipo de público: Niños, Niñas, Adolescentes y Adultos



Parte 3:

siguenos en:    

| Privacidad y Políticas | Términos y Condiciones |

© 2024 | Fundación Trapitos Con Amor | Todos los derechos reservados

Política y Privacidad, Términos y Condiciones

Descripción:

(parte 1) En la parte superior se encuentran los botones que dirigen a las diferentes pestañas. A continuación, se presenta una segunda portada que destaca el título, el cual cambia dinámicamente los colores de la bandera colombiana, sobre un fondo con la imagen de los fundadores. (Parte 2) En el centro se observa el texto de los artículos, resoluciones y leyes,

acompañado de un enlace que permite abrir un archivo PDF con más información al respecto.

(Parte 3) En el footer, se destacan los tres colores de la bandera colombiana, las redes sociales para que los sigan, y los enlaces a la privacidad, políticas, términos y condiciones.

Tabla 33

Política y privacidad diseño

Parte 1:



Parte 2:

Se establecen los diferentes artículos, resoluciones y leyes que rigen la aplicación web sacados de la Resolución 2239 del 2024.

El artículo 15 consagra que todas las personas tienen derecho a su intimidad personal, familiar y a su buen nombre, debiendo el Estado respetarlos y hacerlos respetar. De igual modo, tienen derecho a conocer, actualizar y rectificar la información que se haya recogido sobre ellas en los bancos de datos y en archivos de entidades públicas y privadas.





El artículo 17 de la Ley Estatutaria 1581 de 2012, “Régimen General de Protección de Datos Personales”, y el artículo 2.2.2.25.3.1. del Decreto 1074 de 2015, “Decreto Único Reglamentario del Sector Comercio Industria y Turismo”, consagraron la necesidad de garantizar de forma integral la protección y el ejercicio del derecho fundamental de Habeas Data y estableció dentro de los deberes de los responsables del tratamiento de datos personales, desarrollar políticas para este derecho.

La Ley 1712 de 2014, sobre transparencia y derecho de acceso a la información pública nacional, adiciona nuevos principios, conceptos y procedimientos para el ejercicio y garantía del referido derecho; junto con lo dispuesto en el Libro 2. Parte VIII, Título IV "Gestión de la

ARTÍCULO 6. Política de control de acceso. Los propietarios de los activos de información, teniendo en cuenta el tipo de activo, deberán establecer medidas de control de acceso a nivel de red, sistema operativo, sistemas de información, servicios de tecnologías (on premise o en nube) e infraestructura física (instalaciones y oficinas), todo esto con el fin de mitigar riesgos asociados al acceso a la información, infraestructura tecnológica e infraestructura física de personal no autorizado y así propender por salvaguardar la integridad, disponibilidad y confidencialidad de la información del Ministerio de Tecnologías de la Información y las Comunicaciones.

[Entre otras que contiene este pagina, de le clic para verlas](#)

Parte 3:

siguenos en:    

| [Privacidad y Políticas](#) | [Terminos y Condiciones](#) |

© 2024 | Fundación Trapitos Con Amor | Todos los derechos reservados

Base de datos

Contraseña y usuario

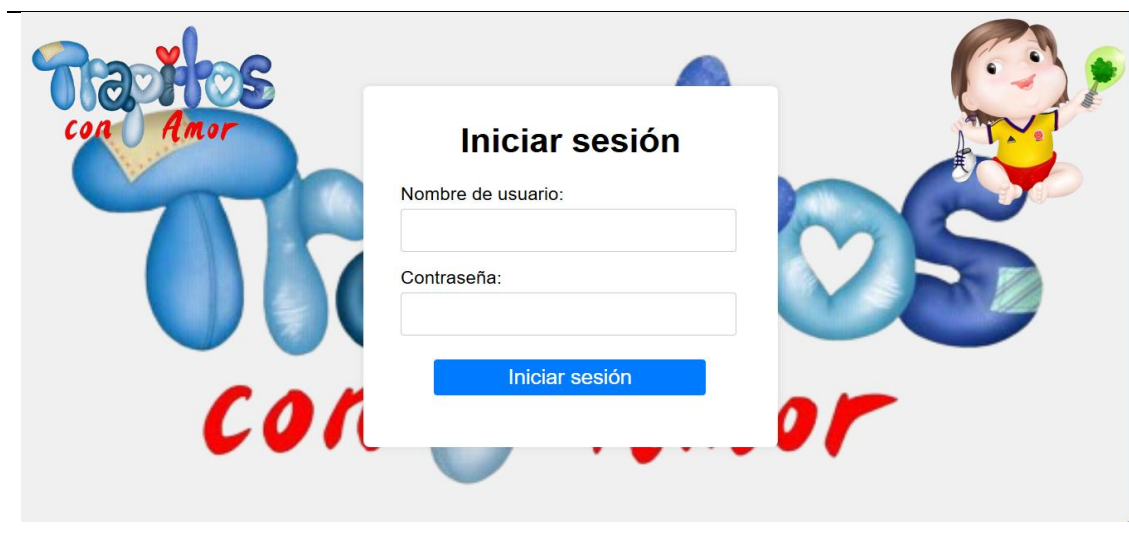
Descripción:

(parte 1) En la parte superior se encuentran los dos logos alusivos a la fundación. En el fondo de la página, hay una imagen con el nombre de la fundación. En el centro, se muestra una interfaz que incluye un título, dos campos para ingresar el nombre y la contraseña, y un botón para acceder a la siguiente sección.

Tabla 34

Contraseña y usuario

Parte 1:



Diseño Tabla

Descripción:

(Parte 1) En la parte izquierda se pueden ver cuatro líneas horizontales de color rojo que, al hacer clic, despliegan el menú. En el centro, se encuentra el nombre de la fundación, y en la parte derecha, el logo de la fundación. **(Parte 2)** Aquí se observa un título denominado "Base de Datos" y siete botones; los primeros seis redirigen a las diferentes tablas correspondientes al nombre de cada botón, mientras que el séptimo botón permite cerrar la sesión. **(Parte 3)** En la parte superior se muestra el nombre de la tabla, en el centro se pueden visualizar los datos ingresados en la tabla de voluntarios altruistas, en cada tabla se podrá ver que contiene sus propios datos. **(Parte 4)** Hay tres cuadros que corresponden a formularios para actualizar, insertar y borrar. Cada formulario incluye campos de texto y un botón.

Tabla 35

Diseño de tablas base de datos

Parte 1:



Parte 2:

Parte 3:



Base de datos de Voluntarios Altruistas:

ID	Username	Teléfono	Departamento	Municipio	Dias/Meses	Motivo	Actividad	Fecha Último
1	Nello mazzotti	3146678990	Cundinamarca	silvania	34 meses	cosa	marisa	2024-08-11T03:01:13.000Z
2	Javier Humberto Santos Poveda	32452627889	Amazonas	Leticia	12 días	me gusta	cantar	2024-08-11T03:01:13.000Z
3	Johan David	17888190100	Antioquia	Medellin	12	cadafha	lalal	2024-08-11T03:01:26.000Z

Insertar Nuevo Voluntario

Username:

Teléfono:

Departamento:

Actualizar Voluntario

ID:

Username:

Teléfono:

Borrar Voluntario

ID:

Parte 4:

Insertar Nuevo Voluntario

Username:

Teléfono:

Departamento:

Municipio:

Días/Meses:

Motivo:

Actividad:

Actualizar Voluntario

ID:

Username:

Teléfono:

Departamento:

Municipio:

Días/Meses:

Motivo:

Actividad:

Borrar Voluntario

ID:

Software

Contraseña y usuario

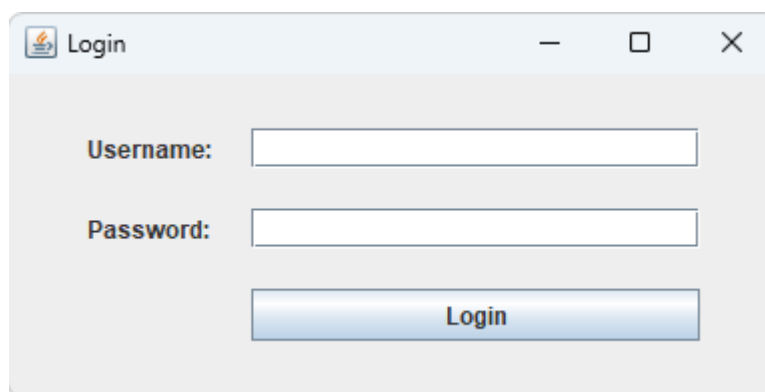
Descripción:

(parte 1) Se observa una interfaz diseñada para ingresar el usuario y la contraseña, con un botón llamado 'Login' que, al ser presionado, mostrará un mensaje de éxito en caso de ingreso correcto o un mensaje de error si los datos son incorrectos.

Tabla 36

Contraseña y usuario Software

Parte 1:



Diseño Tabla

Descripción:

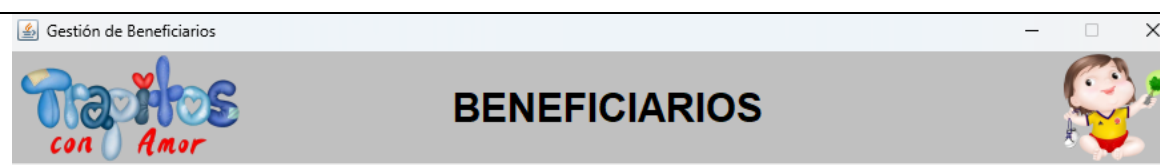
(parte 1) En la parte superior de la interfaz se encuentra una barra de título con botones para maximizar, minimizar y cerrar. Por otro lado, en la parte derecha se encuentra un logotipo que representa el nombre de la fundación, mientras que en la izquierda se ve otro logotipo con la imagen de un bebé que simboliza la fundación, y en el centro se muestra el nombre de la interfaz.

(parte 2) En el centro de la interfaz se muestra una tabla diseñada para presentar el ID, nombre de usuario, identificación, lugar de residencia, lugar de origen, cantidad de beneficios, teléfono y fecha de los usuarios que reciben los beneficios de la fundación. **(parte 3)** En la parte inferior de la interfaz se encuentran tres formularios para insertar, actualizar, buscar y borrar, cada uno con sus respectivos campos, textos y botones.

Tabla 37

Diseño Tabla Software

Parte 1:



Parte 2:

ID	Username	Identification	Lugar Residencia	Lugar Origen
2	Rocio Santos	1074819919	Chipaque	Caqueza
3	Javier Santos	103030030	Une	Une
4	Gloria Hortua	10627828299	Une	Bogota
5	Patricia Hortua	1030308	Chipaque	Italia

ID	Username	Identification	Lugar Residencia	Lugar Origen	Cantidad B.	Telefono	Fecha

Parte 3:

INSERTAR DATO	ACTUALIZAR DATO	
Username: <input type="text"/>	ID a actualizar: <input type="text"/>	<input type="button" value="Reiniciar"/>
Identification: <input type="text"/>	Nuevo Username: <input type="text"/>	BUSCAR
Lugar Residencia: <input type="text"/>	Nueva Identification: <input type="text"/>	Username <input type="text"/>
Lugar Origen: <input type="text"/>	Nuevo Lugar Residencia: <input type="text"/>	<input type="button" value="Buscar"/>
Cantidad Beneficiarios: <input type="text"/>	Nuevo Lugar Origen: <input type="text"/>	BORRAR DATO
Telefono: <input type="text"/>	Nueva Cantidad Beneficiarios: <input type="text"/>	ID a eliminar <input type="text"/>
<input type="button" value="Guardar"/>	Nuevo Telefono: <input type="text"/>	<input type="button" value="Eliminar"/>
	<input type="button" value="Actualizar"/>	

Plan de Pruebas

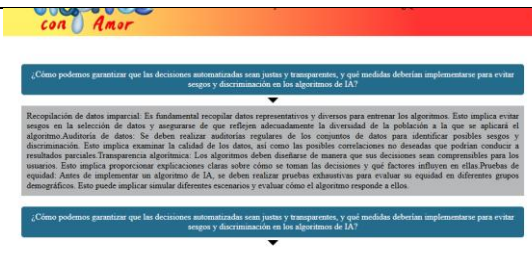
Descripción:

Aquí se mostrará, las diferentes pruebas que se hicieron en el desarrollo y diseño del aplicativo, base de datos y software.

Prueba de pregunta y respuesta

Tabla 38

Prueba de pregunta y respuesta

Prueba 1: Imagen De Lo Que Realice	Prueba 2: En La Que La Segunda
	Respuesta No Deja Visibilizarse
 <p>¿Cómo podemos garantizar que las decisiones automatizadas sean justas y transparentes, y qué medidas deberían implementarse para evitar sesgos y discriminación en los algoritmos de IA?</p> <p>Recopilación de datos imparcial. Es fundamental recopilar datos representativos y diversos para entrenar los algoritmos. Esto implica evitar sesgos en la selección de datos y asegurarse de que reflejen adecuadamente la diversidad de la población a la que se aplicará el algoritmo. Auditoría de datos. Se deben realizar auditorías regulares de los conjuntos de datos para identificar posibles sesgos y discriminación. Esto implica examinar la calidad de los datos, así como las posibles correlaciones no deseadas que podrían conducir a resultados parciales. Transparencia algorítmica. Los algoritmos deben diseñarse de manera que sus decisiones sean comprensibles para los usuarios. Esto implica proporcionar explicaciones claras sobre cómo se toman las decisiones y qué factores influyen en ellas. Pruebas de equidad. Antes de implementar un algoritmo de IA, se deben realizar pruebas exhaustivas para evaluar su equidad en diferentes grupos demográficos. Esto puede implicar simular diferentes escenarios y evaluar cómo el algoritmo responde a ellos.</p> <p>¿Cómo podemos garantizar que las decisiones automatizadas sean justas y transparentes, y qué medidas deberían implementarse para evitar sesgos y discriminación en los algoritmos de IA?</p>	<pre>function ShowHide() { var container = document.getElementsByClassName("respuesta")[0]; if (container.style.visibility == "hidden") { container.style.visibility = "visible"; } else{ container.style.visibility = "hidden"; } };</pre>
Prueba 3: Ingresa La Respuesta Dentro	Prueba 4: Corrección El Javascript, Para
De La Pregunta	Que Funcione Correctamente
<pre><div class= "pregunta"> <button class="btnshow">¿Cómo podemos garantizar que las decisiones automatizadas sean justas y transparentes, y qué medidas deberían implementarse para evitar sesgos y discriminación en los algoritmos de IA? </div></ </button> <div class= "respuesta"> <p class="resp">Recopilación de datos imparcial: Es fundamental recopilar datos representativos y diversos para entrenar los algoritmos.</p> </div> </div></pre>	<pre>document.addEventListener("DOMContentLoaded", function() { var botones = document.querySelectorAll(".btnshow"); botones.forEach(function(boton) { boton.addEventListener("click", function() { var respuesta = this.nextElementSibling; respuesta.classList.toggle("mostrar"); }); }); });</pre>

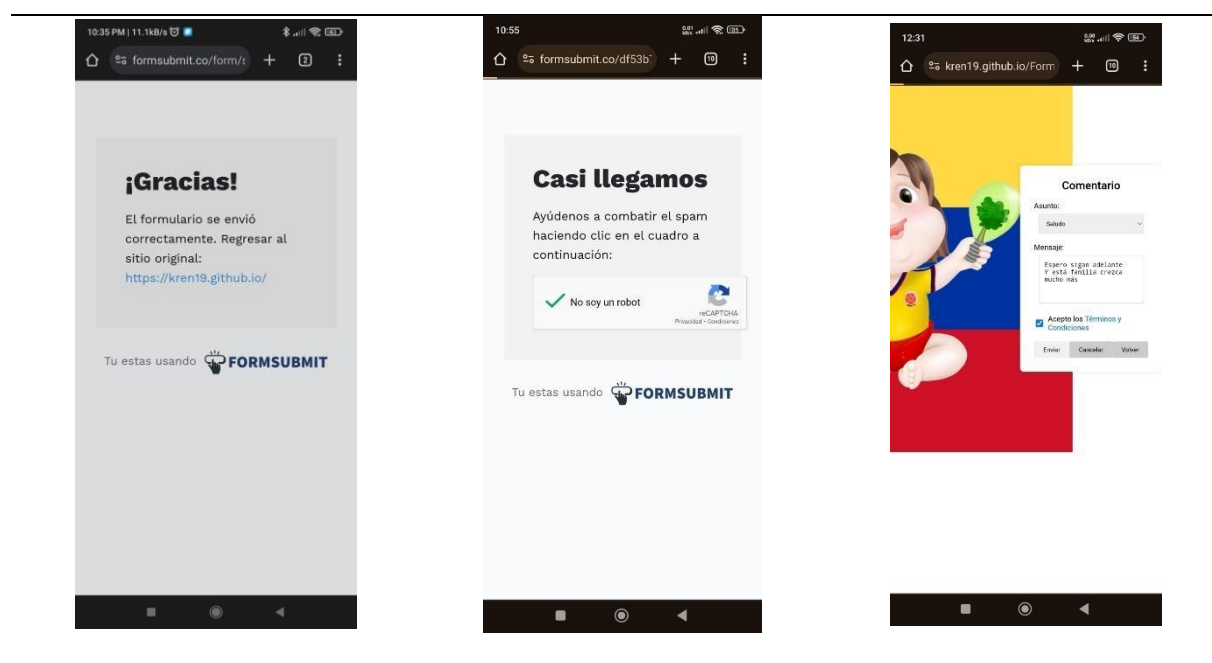
Prueba de formulario por formsubmit

Se realizaron las pruebas del formulario se ajustó de muchas maneras, pero a lo último se usó una página que permite enviar al email, al principio había puesto la css aparte y html,

entonces esto permite cargar con lentitud, dependiendo de la velocidad del internet también afectará, con esto entonces se juntó el css y html, también se ajustó para un dispositivo móvil. Concluyendo también se realiza con el cronómetro la demora, para la computadora duró casi 13 segundos sin los ajustes anteriores y con los ajustes de 8 segundos a 10 segundos. Se hacen pruebas en 3 dispositivos: en un Xiaomi note 8 de 128gb y 4 ram de 5 segundos el envío, Samsung Galaxy S5 de 16gb y 2gb de ram se demoró 25 segundos.

Tabla 39

Prueba Formulario



Instalación del servidor node.js.

Tabla 40

Instalación del servidor

Parte 1:

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\Karen Hortua> npm init -y
Wrote to C:\Users\Karen Hortua\package.json:

{
  "name": "karen-hortua",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \\Error: no test specified\\ && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

PS C:\Users\Karen Hortua> npm install nodemailer express body-parser
added 65 packages, and audited 66 packages in 5s

12 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
PS C:\Users\Karen Hortua> npm install nodemon dotenv
added 30 packages, and audited 96 packages in 6s

17 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
PS C:\Users\Karen Hortua>

```

Parte 2:

```

Windows PowerShell

"version": "1.0.0",
"main": "index.js",
"scripts": {
  "test": "echo \\Error: no test specified\\ && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC",
"description": ""
}

PS C:\Users\Karen Hortua> npm install nodemailer express body-parser
added 65 packages, and audited 66 packages in 5s

12 packages are looking for funding
  run 'npm fund' for details

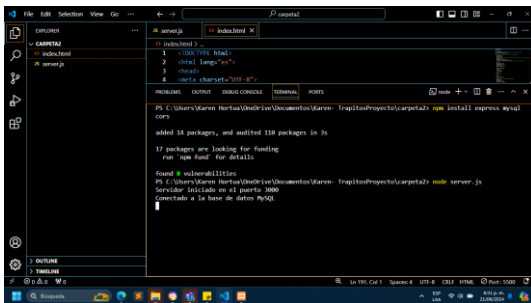
found 0 vulnerabilities
PS C:\Users\Karen Hortua> npm install nodemon dotenv
added 30 packages, and audited 96 packages in 6s

17 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
PS C:\Users\Karen Hortua>

```

Parte 3:



Parte 4:

```

documentos\Karen- TrapitosProyecto\Basededatos-Fundacion> npm init -y
Wrote to C:\Users\Karen Hortua\OneDrive\Documents\Karen- TrapitosProyecto\Basededatos-Fundacion\
package.json:

{
  "name": "basededatos-fundacion",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \\Error: no test specified\\ && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

```

Parte 5:

```

PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- TrapitosProyecto\Basededatos-Fundacion> npm install argon2
added 4 packages, and audited 151 packages in 18s

15 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

```

Parte 6:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

)

PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- TrapitosProyecto\Basededatos-Fundacion> npm install express body-parser bcryptjs mysql
added 77 packages, and audited 78 packages in 38s

12 packages are looking for funding
  run 'npm fund' for details

PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- TrapitosProyecto\Basededatos-Fundacion> npm install express body-parser bcryptjs mysql
added 77 packages, and audited 78 packages in 38s

12 packages are looking for funding
  run 'npm fund' for details

added 77 packages, and audited 78 packages in 38s

12 packages are looking for funding
  run 'npm fund' for details

12 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- TrapitosProyecto\Basededatos-Fundacion >

```

Parte 7:

Parte 8:

```
PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- Trapijos\Proyecto\Basedatos-Fundacion> npm install express body-parser
up to date, audited 78 packages in 3s
12 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- Trapijos\Proyecto\Basedatos-Fundacion> node server.js
```

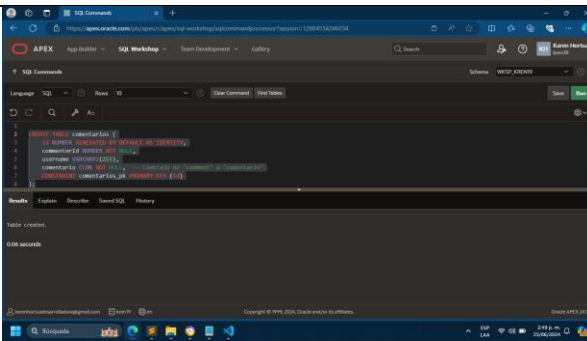
```
PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- Trapijos\Proyecto\Basedatos-Fundacion> npm install bcrypt
npm warn deprecated @jridgway/pretty-print-ast@1.0.0: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way t
0 coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v8 are no longer supported
npm warn deprecated eslint@6.8.1: This package is no longer supported.
npm warn deprecated rimraf@3.0.2: rimraf versions prior to v4 are no longer supported
npm warn deprecated am-unique-id@1.0.0: This package is no longer supported.
npm warn deprecated gauge@3.0.2: This package is no longer supported.
added 57 packages, and audited 135 packages in 10s
16 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\Karen Hortua\OneDrive\Documents\Karen- Trapijos\Proyecto\Basedatos-Fundacion>
```

Prueba de subida de una parte del diseño y adjuntar la base de datos en linea.

Tabla 41

Prueba diseño y adjuntar la base de datos

Parte 1:



Parte 2:

```
git add gitattributes
git commit -m "add gitattributes to handle line endings"
git push origin main
git add --renormalize
git commit -m "normalize all the line endings"
git push origin main
git status
git push origin main
git add
git commit -m "initial commit"
git push origin main
git add
git commit -m "initial commit"
git push origin main
git add
git commit -m "initial commit"
git push origin main
```

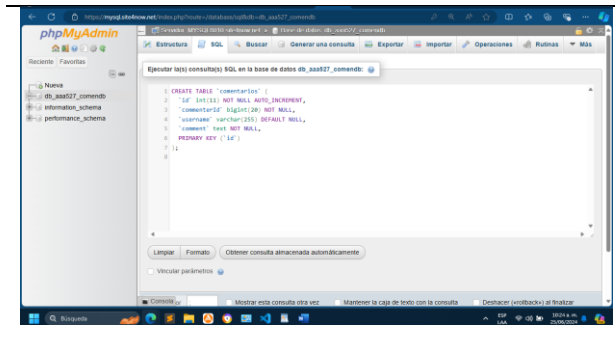
Parte 3:

```
git init
git remote add origin https://github.com/KarenHortua/comentarios-frontend.git
git push origin main
git remote add origin https://github.com/KarenHortua/comentarios-backend.git
git push origin main
git add
git commit -m "Initial commit"
git push origin main
```

Parte 4:

```
git init
git remote add origin https://github.com/KarenHortua/comentarios-frontend.git
git push origin main
git remote add origin https://github.com/KarenHortua/comentarios-backend.git
git push origin main
git add
git commit -m "Initial commit"
git push origin main
```

Parte 5:



Pestaña Donar

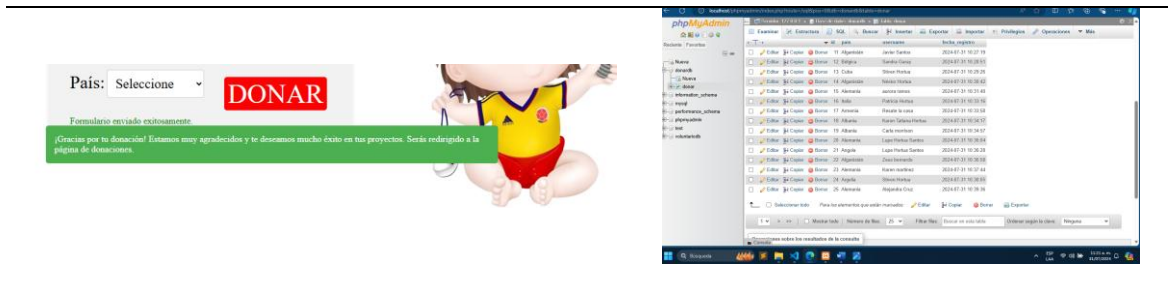
La columna published, Caso 'POST', Este caso maneja la inserción de un nuevo comentario. Los datos del comentario se reciben del cuerpo de la solicitud POST en formato JSON. Se inserta un nuevo registro en la tabla comentarios con los campos commenterId, username, comment and published. El campo publicado se establece en 0 por defecto, lo que significa que el comentario no se publica inmediatamente. Se devuelve una respuesta JSON que incluye el ID del comentario insertado si la inserción fue exitosa, o un mensaje de error si hubo algún problema.

Tabla 42

Prueba pestaña Donar

Parte 1:

Parte 2:



Conclusiones

Descripción:

En el proyecto se realizó la planificación y levantamiento de requerimientos funcionales, apoyados en distintas formas de trabajo con fundación, como llamadas telefónicas y asistencia presencial a la sede. Esto permitió visualizar el diseño a implementar, ajustado a las necesidades específicas del cliente.

Como resultado, se diseñó la arquitectura del software basada en los requerimientos obtenidos, proponiendo la mejor solución para la infraestructura TI existente.

Durante este proceso, profundicé en las diferencias fundamentales entre el **frontend** y el **backend**, dos componentes esenciales en el desarrollo web. Comprender sus roles y formas de interacción me permitió mejorar significativamente mi enfoque de programación, optimizando tanto el diseño como la funcionalidad de las aplicaciones.

Asimismo, enfrenté diversos errores durante el desarrollo, desde fallas en el código hasta desafíos de integración e infraestructura. Aunque en un inicio parecían obstáculos complejos, pude resolverlos aplicando conocimientos adquiridos sobre Inteligencia Artificial. Esto no solo me ayudó a superar los problemas, sino también a obtener aprendizajes valiosos que enriquecieron mi proceso de desarrollo.

Aprendí a organizar cada pestaña —digital o física— según un orden lógico transformó mi forma de trabajar. Entendí la importancia de una estructura coherente para gestionar tareas y asegurar un flujo de trabajo eficiente.

Como parte del proyecto, también elaboré un plan de mantenimiento del aplicativo web, con el objetivo de corregir errores, mantenerlo actualizado y asegurar la disponibilidad constante de herramientas de seguridad del sistema.

Una de las áreas más fascinantes que descubrí fue el uso de servidores en aplicaciones web. Inicialmente, pensaba que eran solo sistemas de almacenamiento, pero luego comprendí su rol crítico como pilar que permite la interacción entre usuarios y sistemas, así como la gestión eficiente de datos.

Finalmente, exploré y apliqué diversas configuraciones cibernéticas que contribuyeron a mejorar tanto el rendimiento como la seguridad de aplicaciones, software y bases de datos. Esta experiencia me permitió valorar la importancia de una infraestructura tecnológica bien configurada para garantizar el éxito de un proyecto o la eficiencia operativa de una empresa.

Conceptos

Aislamiento (Isolation):

Se ejecutan como si fueran secuenciales. Esto significa que las operaciones de una transacción no deben ser visibles para otras transacciones hasta que se completen.

AJAX:

Es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor.

Arquitectura Monolítica:

Un estilo de diseño de software en el cual todas las funcionalidades de una aplicación se combinan en un único programa o módulo que se ejecuta como un solo proceso. Este enfoque ha sido tradicional en el desarrollo de aplicaciones, especialmente antes de la popularización de los microservicios y otras arquitecturas más distribuidas.

Atomicidad (Atomicity):

Se trata como una unidad indivisible. Esto significa que una transacción se completa en su totalidad o no se realiza en absoluto. En otras palabras, una transacción no puede quedar en un estado intermedio.

Base de Datos (MySQL):

Un sistema de gestión de bases de datos relacional que almacena información estructurada.

Base de datos relacionales:

Es un sistema de gestión de bases de datos que se basa en el modelo relacional. Este modelo organiza los datos en una serie de tablas relacionadas entre sí mediante claves primarias y foráneas.

Consistencia (Consistency):

Esto significa que cualquier transacción ejecutada en la base de datos debe cumplir todas las reglas definidas, como restricciones, cascadas y disparadores.

Datos no estructurados:

Son aquellos que no siguen un formato predefinido o un modelo organizacional establecido, lo que los hace difíciles de almacenar, buscar y analizar usando métodos tradicionales de bases de datos relacionales.

Donantes:

Personas que contribuyen económicamente para apoyar los proyectos y actividades de la fundación.

Durabilidad (Durability):

Se ha completado con éxito, sus cambios son permanentes y se registran en la base de datos, incluso si ocurre un fallo del sistema inmediatamente después.

Gestión de Datos (PHP + MySQL en XAMPP):

Aplicaciones y scripts que manejan la entrada, actualización y consulta de datos en la base de datos.

Hash:

El hash es lo que se almacenará en la base de datos en lugar de la contraseña en texto plano. Si ocurre algún error, se captura y se imprime.

Node.js:

Un entorno de ejecución de JavaScript que permite ejecutar código JavaScript en el servidor, fuera del navegador. Fue desarrollado en 2009 por Ryan Dahl y se basa en el motor V8 de Google, el mismo que se utiliza en el navegador Chrome para interpretar y ejecutar JavaScript.

Pasarelas de pago:

Componentes que integran sistemas de pago y logística para facilitar transacciones.

POST:

Se utiliza cuando el cliente desea enviar datos al servidor, normalmente para la creación o actualización de recursos. A diferencia de las operaciones GET, las solicitudes POST pueden alterar el estado del servidor, ya que pueden crear, modificar o eliminar registros en la base de datos.

Transacciones ACID:

Es un conjunto de propiedades garantizadas para las transacciones en sistemas de gestión de bases de datos, asegurando la integridad de los datos a pesar de errores, fallos y otros problemas.

Visitantes:

Personas que visitan el sitio web para obtener información, conocer más sobre la fundación o interesarse en participar en sus actividades.

Voluntarios:

Personas que ofrecen su tiempo y habilidades para apoyar las actividades de la fundación.

Referencias Bibliográficas

A., D. (2024). Cómo usar Hostinger para tener éxito en Internet.

<https://www.hostinger.co/tutoriales/como-usar-hostinger>

Cdnjs. (2024). font-awesome. <https://cdnjs.com/libraries/font-awesome>

Chris, K. (8 de julio de 2022). Etiqueta select de HTML: Cómo hacer un menú desplegable o lista combinada. <https://www.freecodecamp.org/espanol/news/etiqueta-select-de-html-como-hacer-un-menu-desplegable-o-lista-combinada/>

Domingo, F. S. (2024). Fundación Santo Domingo.

https://fundacionsantodomingo.org/?gad_source=1&gclid=Cj0KCQjwpNuyBhCuARIsANJqL9PRDasKH0rTZqZU-eiO0vkzAdwNa1RbyM5UEQtjz31yUB3Aj23DS-EaA15zEALw_wcB

Font-awesome. (s.f., 2024). ICONOS PARA PAGINAS Y APLICACIONES WEB.

<https://fontawesome.com/icons>

García, M. Á. (s.f., 2024). Formsubmit.

<https://formsubmit.co/>

OpenAI. (2023). ChatGPT.

<https://www.openai.com/chatgpt>

Percona. (s.f., 2024). <https://www.percona.com/blog/mysql-database-security-best-practices/>

Vélez, F. (2020). Fundación Vélez. https://fundacionvelez.org/?gad_source=1#tabs-1?utm_source=google_search&utm_medium=18484444978&utm_campaign=145624963367&utm_content=625511469118&utm_term=fundaciones%20que%20reciben%20donaciones&gclid=Cj0KCQjwpNuyBhCuARIsANJqL9OmeRnZRn8VNtTz4rhzugRol_Y9jZpVGGABw57S7deM_Y54WmsZFEAaAnPfeALw_wcB

Welling, L., & Thompson, L. (2016). PHP and MySQL Web Development. Addison-Wesley Professional. <https://openlibrary.org/>