

## PORTADA

## **ÍNDICE**

### **1. INTRODUCCION**

### **2. ESQUEMAS Y GRÁFICOS**

### **3. PROGRAMACION DEL PROYECTO**

#### **3.1. Fase A: Planificación y Análisis**

#### **3.2. Fase B: Diseño**

#### **3.3. Fase C: Desarrollo**

#### **3.4. Fase D: Pruebas y Depuración**

#### **3.5. Fase E: Despliegue y Entrega**

### **4. DESARROLLO DE LOS CONTENIDOS**

#### **4.1. Plantilla**

#### **4.2. Login y logout. Acceso a la sesión**

#### **4.3. Gestión de Usuarios**

#### **4.4. Gestión de Proyectos**

#### **4.5. Gestión de Tareas**

#### **4.6. Gestión de Notas**

**4.7. Información adicional**

**5. JUSTIFICACION DE LA VIABILIDAD**

**5.1. Viabilidad Técnica: Tecnología Probada y Arquitectura Modular**

**5.2. Viabilidad Económica: Presupuesto Adecuado**

**5.3. Viabilidad Operativa: Gestión de Proyectos Eficaz**

**5.4. Viabilidad Legal y Regulatoria: Licencias y Permisos**

**6. ASPECTOS TRANSVERSALES DE PREVENCIÓN DE RIESGOS LABORALES**

**6.1. Evaluación Inicial de Riesgos**

**6.2. Medidas de Prevención y Protección**

**6.3. Capacitación y Concienciación**

## **1. INTRODUCCIÓN**

Este proyecto web está diseñado con una arquitectura modular que organiza su funcionalidad en varios componentes PHP. El sitio se compone de varias secciones que trabajan juntas para ofrecer una experiencia de usuario coherente y eficiente.

La estructura del sitio incluye un encabezado HTML y una barra de navegación que definen el estilo y la estructura de la página, incorporando librerías de Bootstrap para una presentación visual atractiva y responsiva. La barra de navegación proporciona enlaces a diferentes secciones del sitio como "Inicio", "Quiénes somos", "Política de privacidad" y "Contacto".

La página principal del sitio muestra información general sobre cómo administrar proyectos, crear tareas, trabajar en equipo y visualizar estadísticas. Esta página maneja sesiones de usuario para personalizar la experiencia y organizar el contenido de manera estructurada.

El pie de página incluye un área de autenticación que muestra un saludo y la funcionalidad de cierre de sesión (destruye la sesión actual y redirige al usuario a la página de inicio de sesión) si el usuario está autenticado, o un formulario de inicio de sesión si no lo está. Una vez autenticado, proporciona enlaces a las funcionalidades principales.

El registro de nuevos usuarios incluye un formulario para ingresar datos personales y de contacto, validando y almacenando la información en la base de datos con contraseñas encriptadas. Además, se proporciona una funcionalidad para la

recuperación de contraseñas.

El sistema de gestión de usuarios se presenta mediante un histórico que permite crear, ver, editar y borrar usuarios. Los administradores pueden ver y filtrar la lista de usuarios, acceder a detalles específicos, crear nuevos usuarios, actualizar datos existentes y borrar usuarios, gestionando los roles y permisos de acceso.

La gestión de proyectos se presenta mediante un histórico que permite crear, ver, editar y borrar proyectos. También permite la acción de asignar usuarios, crear y ver tareas asociadas.

La gestión de tareas se presenta mediante un histórico que permite crear, ver, editar y borrar tareas de proyectos. También permite la acción de asignar usuarios, crear y ver notas asociadas.

Las notas se presentan mediante un histórico que permite crear, ver, editar y borrar, proporcionando una manera de agregar y gestionar comentarios y observaciones relacionadas.

*This web project is designed with a modular architecture that organizes its functionality in several PHP components. The site is made up of several sections that work together to offer a consistent and efficient user experience.*

*The site structure includes an HTML header and navigation bar that define the style and structure of the page, incorporating Bootstrap libraries for an attractive and*

*responsive visual presentation. The navigation bar provides links to different sections of the site such as "Home", "About Us", "Privacy Policy" and "Contact Us".*

*The main page of the site displays general information on how to manage projects, create tasks, work as a team, and view statistics. This page handles user sessions to personalize the experience and organize content in a structured way.*

*The footer includes an authentication area that displays a greeting and logout functionality (destroys the current session and redirects the user to the login page) if the user is authenticated, or a login form if it is not. Once authenticated, it provides links to core functionalities.*

*New user registration includes a form to enter personal and contact data, validating and storing the information in the database with encrypted passwords. Additionally, a password recovery functionality is provided.*

*The user management system is presented through a history that allows users to be created, viewed, edited and deleted. Administrators can view and filter the list of users, access specific details, create new users, update existing data and delete users, managing roles and access permissions.*

*Project management is presented through a history that allows you to create, view, edit and delete projects. It also allows the action of assigning users, creating and viewing associated tasks.*

*Task management is presented through a history that allows you to create, view, edit and delete project tasks. It also allows the action of assigning users, creating and viewing associated notes.*

*Notes are presented through a history that allows you to create, view, edit and delete, providing a way to add and manage comments and related observations.*

## **2. ESQUEMAS Y GRÁFICOS**



### **3. PROGRAMACION DEL PROYECTO**

#### **3.1. Fase A: Planificación y Análisis**

Definir los requisitos del proyecto: Documentar los requisitos funcionales y no funcionales del sitio.

Identificar los componentes y funcionalidades del sitio: Esbozar la estructura del sitio e identificando los módulos clave.

#### **3.2. Fase B: Diseño**

Crear el diseño visual y de interacción del sitio: Diseño de la interfaz de usuario para las páginas principales del sitio.

Desarrollar prototipos: Desarrollar prototipos interactivos para probar la navegación y la usabilidad.

#### **3.3. Fase C: Desarrollo**

Configuración de la DB: Crear la base de datos y establecer relaciones entre los diferentes campos.

Implementar la estructura modular del sitio web: Implementar el diseño y funcionalidad de la cabecera, así como la navegación. Crear la página principal con contenido dinámico y paginación. Desarrollar el pie de página y la lógica de inicio/cierre de sesión.

Desarrollar y probar cada módulo del sitio: Crear las funcionalidades para la creación, lectura, actualización y borrado de usuarios (CRUD). Desarrollar las funcionalidades de proyectos, tareas y notas (CRUD) y asignaciones de usuarios.

### **3.4. Fase D: Pruebas y Depuración**

Asegurar la calidad y funcionalidad del sitio web: Probar individualmente cada módulo para asegurar su correcta funcionalidad. Asegurar que los módulos funcionan correctamente juntos. Realizar pruebas con diferentes usuarios para validar la usabilidad y experiencia.

Corregir errores y optimizar el rendimiento: Corregir errores encontrados y optimizar el código. Asegurar la implementación del comportamiento correcto en todas las casuísticas.

### **3.5. Fase E: Despliegue y Entrega**

Implementar el sitio web en el entorno de producción: Configurar servidores y bases de datos en producción. Migrar el código y la base de datos al entorno de

producción.

Asegurar la correcta operación del sitio web en el entorno de producción: Realizar pruebas finales para asegurar que todo funcione correctamente. Entregar el proyecto y proporcionar la documentación necesaria.

## **4. DESARROLLO DE LOS CONTENIDOS**

### **4.1. Plantilla:**

La plantilla del sitio web está diseñada utilizando una arquitectura modular que facilita la organización y reutilización de los componentes. Los scripts que constituyen esta plantilla son header.php, index.php y footer.php. Estos archivos se integran de manera coherente para formar la estructura del sitio web y asegurar una experiencia de usuario fluida y consistente.

El archivo header.php contiene el encabezado HTML y la barra de navegación del sitio. Este script se encarga de incluir las librerías de Bootstrap, definir el estilo CSS y configurar la estructura básica del documento HTML, así como los metadatos necesarios para el correcto funcionamiento y visualización del sitio web. La barra de navegación incluye elementos como el logo del sitio, el título, y enlaces de navegación que permiten a los usuarios acceder a diferentes secciones del sitio web como "Inicio", "Quiénes somos", "Política de privacidad" y "Contacto".

El archivo index.php actúa como la página principal del sitio web. Su principal responsabilidad es mostrar el contenido principal y garantizar que tanto header.php como footer.php se incluyan adecuadamente. Al comienzo del archivo, index.php utiliza `session_start()` para gestionar las sesiones de usuario, lo que es crucial para funcionalidades como el inicio de sesión y la personalización del contenido. La inclusión de header.php al inicio y footer.php al final del archivo permite mantener una consistencia en el diseño y la navegación en todas las páginas del sitio web.

El archivo footer.php completa la plantilla al proporcionar el pie de página del sitio web y un área lateral derecha dedicada a la autenticación de usuarios. Si un usuario está autenticado, footer.php muestra un saludo personalizado; si no, se presenta un formulario de inicio de sesión. Además, el pie de página ofrece un menú de usuario con enlaces a funcionalidades importantes como la gestión de usuarios, proyectos y estadísticas, visibles solo para los usuarios autenticados. El pie de página también incluye enlaces a otras secciones del sitio web y muestra información adicional sobre la empresa. Finalmente, este archivo incorpora los scripts necesarios para el correcto funcionamiento de Bootstrap y otras funcionalidades JavaScript, asegurando que todos los elementos interactivos del sitio operen sin problemas.

En conjunto, estos tres archivos trabajan de manera integrada para formar una estructura cohesiva y eficiente para el sitio web. La interrelación entre estos archivos es crucial para la modularidad del proyecto. Al mantener el encabezado y el pie de página en archivos separados, cualquier cambio en el diseño o la funcionalidad de estos componentes se puede realizar en un solo lugar, reflejándose inmediatamente en todas las páginas del sitio. Esto no solo facilita el mantenimiento y la actualización del sitio web, sino que también asegura una experiencia de usuario consistente. La inclusión de librerías y scripts en garantiza que todas las páginas tengan acceso a los mismos recursos, mejorando así la cohesión del diseño y la eficiencia del desarrollo.

#### **4.2. Login y logout. Acceso a la sesión**

El sistema de gestión de acceso y manejo de sesiones en el sitio web está compuesto por varios scripts que funcionan de manera coordinada para proporcionar una experiencia de usuario fluida y segura.

El archivo login.php maneja la lógica de inicio de sesión de los usuarios. Este script recibe y valida las credenciales ingresadas (DNI y contraseña), verificando no solo la autenticidad de estos datos, sino también el estado del usuario, asegurándose de que esté activo. Si las credenciales son correctas y el usuario está activo, se establecen las variables de sesión necesarias para mantener la sesión del usuario autenticado. En caso de fallos en la autenticación, el script proporciona retroalimentación adecuada, indicando los errores encontrados mediante los correspondientes mensajes.

Por otro lado, logout.php se encarga de gestionar el cierre de sesión. Este script destruye la sesión actual, garantizando que todas las variables de sesión se eliminen, y luego redirige al usuario a la página de inicio de sesión. Esta acción asegura que el acceso no autorizado a las funcionalidades del sitio web sea restringido, manteniendo la seguridad de la información del usuario.

Complementando estos dos scripts, nuevoRegistro.php permite a nuevos usuarios registrarse en el sitio. Este archivo muestra un formulario donde los usuarios pueden ingresar sus datos personales y de contacto. Una vez ingresados los datos, el script realiza una validación y almacena la información en la base de datos, asegurándose de encriptar las contraseñas antes de guardarlas para mantener la seguridad. Si hay algún error en el proceso de registro, se proporciona retroalimentación para que el usuario pueda corregir los datos ingresados.

Además, olvideContrasenya.php facilita la recuperación de contraseñas para los usuarios que las hayan olvidado. Este script muestra un formulario donde los usuarios pueden ingresar su DNI y correo electrónico. Luego, verifica si esta información es correcta y, de ser así, avisa al administrador para que gestione la

recuperación de la contraseña. Este proceso garantiza que los usuarios puedan recuperar el acceso a sus cuentas de manera segura.

Estos scripts funcionan conjuntamente para asegurar que los usuarios puedan acceder al sitio de manera segura y gestionar sus sesiones eficientemente. La estructura modular del proyecto permite que cada funcionalidad esté claramente definida y aislada, facilitando tanto el mantenimiento como la escalabilidad del sistema. Cada archivo tiene una responsabilidad específica, contribuyendo a un sistema de gestión de usuarios robusto y fiable, esencial para el buen funcionamiento del sitio web.

#### **4.3. Gestión de Usuarios**

El sistema de gestión de usuarios en el sitio web se compone de varias funcionalidades que permiten crear, ver, editar y borrar usuarios, facilitando una administración eficiente y segura. Estos procesos están distribuidos en diferentes archivos que trabajan en conjunto para proporcionar una experiencia de usuario coherente y fluida.

El archivo `mostrarUsuarios.php` se encarga de mostrar un histórico de usuarios, aplicando filtros y paginación para mejorar la usabilidad. Dependiendo del rol del usuario (administrador o no), construye una consulta SQL adecuada para obtener los datos correspondientes, aplicando los filtros de búsqueda y ordenación y limitando los resultados según la paginación. Una vez ejecutada la consulta, obtiene los resultados y, en caso de error, muestra un mensaje de alerta. La interfaz de usuario incluye formularios para la búsqueda; los administradores tienen acceso tanto a sus datos como a los de otros usuarios.

El archivo `verUsuario.php` se centra en mostrar los detalles de un usuario específico. El archivo recoge el DNI del usuario a consultar desde los parámetros de la URL y busca los datos en la base de datos. Finalmente, muestra la información detallada del usuario en una tarjeta, incluyendo un botón para volver a la página anterior, lo que facilita la navegación del usuario.

Para la creación de nuevos usuarios, el archivo `nuevoUsuario.php` proporciona un formulario que recoge los datos necesarios. Si se envía el formulario, recoge los datos del nuevo usuario, valida el DNI y comprueba si ya existe en la base de datos. Si los datos son válidos y el DNI no está registrado, encripta la contraseña e inserta los datos en la base de datos. El script también muestra mensajes de éxito o error según el resultado de la operación. La interfaz de usuario presenta un formulario claro y fácil de usar para introducir los datos del nuevo usuario.

El archivo `editarUsuario.php` permite actualizar los datos de un usuario existente. Recoge el DNI del usuario a editar y obtiene sus datos actuales de la base de datos. Si se envía el formulario, recoge los nuevos datos, los valida y actualiza en la base de datos. Si se introduce una nueva contraseña, esta se encripta antes de ser almacenada. El script muestra mensajes de éxito o error según el resultado de la operación. La interfaz de usuario presenta un formulario prellenado con los datos actuales del usuario, facilitando la edición de la información.

Finalmente, `borrarUsuario.php` maneja la eliminación de usuarios. Recoge el DNI del usuario a borrar y obtiene sus datos de la base de datos. Muestra un formulario para confirmar la acción de borrado del usuario. Si se confirma, el script desactiva al usuario (se ha utilizado borrado lógico para mantener la integridad de los datos) de



la base de datos y muestra un mensaje de éxito o error según el resultado de la operación.

En conjunto, estos scripts proporcionan un sistema completo de gestión de usuarios, asegurando que la administración de usuarios sea eficiente, segura y fácil de usar, mejorando la gestión del sistema en su totalidad.

#### **4.4. Gestión de Proyectos**

El sistema de gestión de proyectos se compone de varias funcionalidades que permiten crear, ver, editar y borrar proyectos, así como la asignación de usuarios a estos. Estos procesos están distribuidos en diferentes archivos que trabajan en conjunto para proporcionar una experiencia de usuario coherente y fluida.

El archivo `mostrarProyectos.php` se encarga de mostrar un histórico de proyectos, proporcionando funcionalidades de búsqueda, paginación y ordenamiento, además de acciones específicas sobre los proyectos como visualizar, editar, asignar usuarios y crear y ver tareas. Define la configuración de paginación con un límite de proyectos por página y calcula el inicio de los registros a mostrar basándose en la página actual. Para la búsqueda, captura el término de búsqueda ingresado por el usuario y lo utiliza para filtrar los resultados de la consulta SQL. También incluye un botón para crear un nuevo proyecto. El histórico genera una consulta SQL dinámica dependiendo del rol del usuario, obteniendo todos los proyectos si el usuario es administrador o solo los proyectos del usuario autenticado. Prepara y ejecuta la consulta SQL usando PDO, manejando posibles excepciones. Muestra los proyectos en una tabla con varias acciones disponibles para cada proyecto.

El archivo `verProyecto.php` muestra la información detallada de un proyecto específico. Captura el ID del proyecto desde la URL y realiza una consulta SQL para obtener todos los detalles del proyecto usando el ID. Si el proyecto no es encontrado, muestra un mensaje de alerta y redirige al usuario. Realiza una consulta para obtener los nombres de los usuarios asignados al proyecto y muestra un mensaje si no hay usuarios asignados. La interfaz de usuario muestra toda la información del proyecto y los usuarios asignados en un formato detallado.

El archivo `nuevoProyecto.php` gestiona la creación de nuevos proyectos. Proporciona un formulario donde el usuario puede ingresar los detalles del nuevo proyecto. Si se envía el formulario, recupera los datos y los inserta en la base de datos, realizando validaciones básicas como fechas y obligatoriedad de campos. Muestra mensajes de éxito o error dependiendo del resultado de la inserción. Incluye una función de JavaScript para validar las fechas ingresadas, asegurando que la fecha de inicio no sea anterior a la fecha actual y que la fecha de finalización sea posterior a la fecha de inicio.

El archivo `editarProyecto.php` gestiona la actualización de proyectos existentes. Proporciona un formulario con los datos actuales del proyecto precargados, permitiendo al usuario actualizar los detalles del proyecto. Si se envía el formulario, recupera los datos y actualiza la base de datos con los nuevos valores, realizando las mismas validaciones básicas que en la creación. Incluye la misma función de JavaScript para validar las fechas ingresadas.

El archivo `borrarProyecto.php` maneja la eliminación de proyectos. Recoge el ID del proyecto a borrar y obtiene sus datos de la base de datos. Muestra un formulario para confirmar la acción de borrado del proyecto. Si se confirma, el script desactiva

el proyecto (se ha utilizado borrado lógico para mantener la integridad de los datos) de la base de datos y muestra un mensaje de éxito o error según el resultado de la operación.

Finalmente, el archivo `asignarUsuarioAProyecto.php` permite asignar usuarios a un proyecto específico. Muestra dos listas de usuarios: una con usuarios no asignados al proyecto y otra con usuarios ya asignados. Permite mover usuarios entre estas listas, facilitando la gestión de los equipos de trabajo dentro de cada proyecto.

Este CRUD maneja diferentes roles de usuario y proporciona interfaces de usuario intuitivas para cada operación. Las operaciones de bases de datos son seguras y manejadas a través de PDO, e incluyen validaciones tanto del lado del servidor como del cliente para asegurar la integridad de los datos. Esto garantiza que la gestión de proyectos sea eficiente, segura y fácil de usar, mejorando la administración general del sistema.

#### **4.5. Gestión de Tareas**

El sistema de gestión de tareas en este proyecto se compone de varios archivos que manejan diferentes aspectos del ciclo de vida de una tarea, incluyendo crear, ver, editar y borrar tareas, así como la asignación de usuarios a estas. Estos archivos trabajan juntos para proporcionar una experiencia de usuario coherente y funcional.

El archivo `mostrarTareas.php` es responsable de mostrar todas las tareas de un proyecto específico. Verifica si se ha proporcionado el `proyecto_id` mediante `$_GET`. Configura la paginación mostrando cuatro tareas por página y determina la página

actual y el orden de las tareas. Dependiendo del usuario, ejecuta una consulta SQL para obtener las tareas del proyecto, ya sean todas las tareas o solo aquellas asignadas al usuario. Calcula el número total de tareas para la paginación. La interfaz de usuario incluye una tabla HTML para mostrar el histórico de las tareas.

El archivo verTarea.php muestra los detalles de una tarea específica. Realiza varias consultas a la base de datos para obtener la información de la tarea basada en su ID, el nombre del proyecto al que pertenece la tarea, el nombre del usuario asignado (si lo hay) y el nombre del creador de la tarea. La interfaz de usuario muestra los detalles de la tarea en una tarjeta, incluyendo el nombre, descripción, proyecto, usuario asignado, fechas, estado, creador y si está activa o no.

El archivo nuevaTarea.php permite a los usuarios crear una nueva tarea para un proyecto. Recoge los datos del formulario y los inserta en la base de datos, mostrando mensajes de éxito o error y redirigiendo a la página correspondiente. La interfaz de usuario muestra un formulario para la creación de una nueva tarea con campos para el nombre, descripción, fechas, estado y activo. Incluye validaciones en JavaScript para asegurarse de que las fechas son correctas.

El archivo editarTarea.php permite a los usuarios actualizar una tarea existente. Cuando el formulario ha sido enviado para actualizar, recoge los datos del formulario y actualiza la tarea en la base de datos, mostrando mensajes de éxito o error y redirigiendo a la página correspondiente. Realiza una consulta a la base de datos para obtener la información de la tarea basada en su ID para prellenar el formulario de edición. La interfaz de usuario muestra un formulario para la edición de la tarea con campos para el nombre, descripción, fechas, estado y activo. Incluye las mismas validaciones en JavaScript para asegurarse de que las fechas son correctas.

El archivo `borrarTarea.php` maneja la eliminación de tareas. Recoge el ID de la tarea a borrar y obtiene sus datos de la base de datos. Muestra un formulario para confirmar la acción de borrado del tarea. Si se confirma, el script desactiva la tarea (se ha utilizado borrado lógico para mantener la integridad de los datos) de la base de datos y muestra un mensaje de éxito o error según el resultado de la operación.

Finalmente, el archivo `asignarUsuarioATarea.php` permite asignar usuarios a una tarea específica. Muestra dos listas de usuarios: una con usuarios no asignados a la tarea pero que están asignados al proyecto al que pertenece la tarea, y otra con usuarios ya asignados a la tarea. Permite mover usuarios entre estas listas, facilitando la gestión de los equipos de trabajo dentro de cada tarea.

Estos archivos PHP trabajan conjuntamente para gestionar las tareas de un proyecto, asegurando que los usuarios puedan realizar estas operaciones de manera eficiente y segura. La interfaz de usuario está diseñada para ser intuitiva, proporcionando acceso fácil a las funcionalidades esenciales y validaciones para asegurar la integridad de los datos. Esto garantiza que la gestión de tareas dentro del proyecto sea efectiva y organizada.

#### **4.6. Gestión de Notas**

El sistema de gestión de notas se compone de varios scripts que permiten crear, ver, editar y borrar notas, proporcionando una funcionalidad completa para los usuarios. Estos componentes trabaja en conjunto para ofrecer una experiencia de usuario coherente.

El archivo `mostrarNotas.php` presenta un histórico con todas las notas relacionadas con la tarea de la que se trate. Verifica si se ha proporcionado el `proyecto_id` y un `tarea_id` mediante `$_GET`. Configura la paginación mostrando cuatro notas por página y determina la página actual y el orden de las tareas. Calcula el número total de tareas para la paginación.

El archivo `crearNota.php` permite a los usuarios crear una nueva nota. Cuando el formulario de creación de nota es enviado, se recuperan los datos ingresados por el usuario, como el nombre de la nota y su contenido. Se genera automáticamente la fecha de creación y se prepara una consulta SQL para insertar la nueva nota en la base de datos. Si la inserción es exitosa, el usuario es redirigido a la página de notas correspondiente.

El archivo `editarNota.php` permite a los usuarios editar una nota existente. Similar al archivo de creación. Cuando se envía el formulario de edición, se recuperan los datos actualizados de la nota y se prepara una consulta SQL para actualizar la nota en la base de datos. Tras una actualización exitosa, el usuario es redirigido a la página de notas correspondiente. Si el ID de la nota es recibido a través de la URL, se ejecuta una consulta SQL para obtener la información actual de la nota y prellenar el formulario con estos datos. La interfaz de usuario muestra un formulario con los datos actuales de la nota, permitiendo que el usuario los edite según sea necesario.

El archivo `borrarNota.php` permite a los usuarios desactivar (se ha utilizado borrado lógico para mantener la integridad de los datos) una nota. Al recibir un ID de nota a través de la URL, se obtiene la información de la nota para confirmar su borrado. Si el usuario confirma la eliminación mediante el formulario, se actualiza el estado de la nota a inactiva.

En conjunto, estos archivos proporcionan una funcionalidad completa de gestión de notas dentro del sistema. Estos scripts trabajan de manera coordinada para asegurar que los usuarios puedan gestionar sus notas de manera eficiente, con una interfaz de usuario intuitiva y procesos de validación para garantizar la integridad de los datos.

#### **4.7. Información adicional**

El sistema de información adicional incluye varias secciones que proporcionan a los usuarios detalles sobre el proyecto, la política de privacidad y la información de contacto. A continuación, se explica cómo funcionan estos scripts y cómo se integran en el sistema.

El archivo `quienesSomos.php` contiene una sección principal que despliega un mensaje de bienvenida, explicando que el proyecto ha sido creado con fines académicos. Se informa a los usuarios que los datos ingresados no están garantizados de manera permanente y que podrían perderse, indicando la naturaleza temporal y experimental del proyecto. Esto prepara las expectativas de los usuarios respecto a la estabilidad y durabilidad del sistema.

El archivo `politicaDePrivacidad.php` presenta una declaración de política de privacidad a los usuarios, explicando que la información solicitada es mínima y no es necesario ingresar datos reales. Se menciona que cualquier dato ingresado puede ser modificado a través de la gestión de usuarios, asegurando que no persistirá ninguna información previa si el usuario decide cambiarla. Se detalla que la

información del usuario no se utiliza para ningún propósito específico y que no se comparte con terceros. Además, se enfatiza la seguridad del sistema, asegurando que otros usuarios no pueden acceder a la información personal de los demás. La administradora (yo) tiene control exclusivo sobre los datos, reforzando la privacidad y la seguridad.

El archivo contacto.php proporciona la información de contacto del proyecto, como el correo electrónico, el teléfono y la dirección física. Esta información se presenta en un formato de panel, asegurando que sea fácilmente accesible y visible para los usuarios que necesiten ponerse en contacto con el administrador del sistema.

En conjunto, estos archivos proporcionan información adicional a los usuarios del sistema, mejorando la transparencia y la comunicación. La estructura consistente asegura que solo los usuarios autenticados accedan a esta información, manteniendo un entorno seguro y controlado.



## **5. JUSTIFICACION DE LA VIABILIDAD**

La viabilidad del proyecto de desarrollo del sitio web modular está respaldada por varios factores que demuestran su potencial para ser completado con éxito dentro de los límites de tiempo, presupuesto y recursos disponibles.

### **5.1. Viabilidad Técnica: Tecnología Probada y Arquitectura Modular**

El proyecto utiliza tecnologías web bien establecidas y ampliamente adoptadas, como PHP para el desarrollo del backend, HTML, Bootstrap, CSS y JavaScript para el frontend, y MySQL para la gestión de la base de datos. Estas tecnologías son conocidas por su estabilidad, escalabilidad y soporte comunitario robusto, lo que reduce significativamente los riesgos técnicos.

La arquitectura modular del sitio web facilita la gestión del desarrollo y el mantenimiento. Al dividir la funcionalidad en módulos independientes, cada componente del sitio web puede ser desarrollado, probado y desplegado de forma aislada. Esto permite un desarrollo paralelo y una fácil integración de nuevas funcionalidades, mejorando la eficiencia y reduciendo el tiempo de desarrollo.

### **5.2. Viabilidad Económica: Presupuesto Adecuado**

El proyecto ha sido presupuestado de manera realista, considerando todos los costos, incluida la adquisición de un hosting para un despliegue correcto.

### **5.3. Viabilidad Operativa: Gestión de Proyectos Eficaz**

Se ha establecido una planificación detallada del proyecto con fases claras y plazos definidos. Un plan de gestión de proyectos riguroso, respaldado por herramientas de seguimiento y control, asegura que las actividades se realicen según lo programado.

### **5.4. Viabilidad Legal y Regulatoria: Licencias y Permisos**

Todas las herramientas y tecnologías utilizadas en el proyecto cuentan con las licencias y permisos necesarios, garantizando un uso legal y evitando conflictos de propiedad intelectual.

## **6. ASPECTOS TRANSVERSALES DE PREVENCIÓN DE RIESGOS LABORALES**

La prevención de riesgos laborales es un aspecto fundamental en la planificación y ejecución del proyecto de desarrollo del sitio. Asegurar un entorno de trabajo seguro y saludable contribuye a la eficiencia y el éxito del proyecto.

### **6.1. Evaluación Inicial de Riesgos**

Antes del inicio del proyecto, se realizó una evaluación de los riesgos potenciales. Esta evaluación identificó posibles peligros relacionados con el uso de equipos informáticos, la ergonomía, el estrés laboral, y el entorno físico del lugar de trabajo.

## **6.2. Medidas de Prevención y Protección**

**Ergonomía:** Se han implementado medidas ergonómicas para asegurar que el espacio de trabajo sea cómodo y seguro. Esto incluye la provisión de silla ajustable, escritorio adecuado, y la correcta disposición del equipo informático para prevenir lesiones musculoesqueléticas.

**Equipos de Protección Personal (EPP):** Aunque el trabajo de desarrollo web no suele requerir EPP, se han establecido directrices para el uso de equipos de protección adecuados en caso de situaciones específicas, como el manejo de materiales eléctricos o de limpieza.

**Higiene y Seguridad en el Lugar de Trabajo:** Se garantiza un entorno de trabajo limpio y seguro. Esto incluye la limpieza regular de las áreas de trabajo, la disponibilidad de desinfectantes y la implementación de prácticas de higiene adecuadas, especialmente relevante en el contexto post-pandemia.

## **6.3. Capacitación y Concienciación**

**Formación en Prevención de Riesgos:** La formación incluye la identificación de riesgos, el uso correcto de equipos, y procedimientos de emergencia.

**Concienciación sobre la Salud Mental:** Reconociendo el impacto del estrés laboral,

se promueven prácticas de trabajo saludables, como pausas regulares o balance entre trabajo y vida personal.