**Introducción**

Somos un equipo de estudiantes del Colegio Secundario “Alte. Brown” y hemos decidido emprender un proyecto que tiene como objetivo la creación de una página web dedicada a la asistencia en reparaciones y reconstrucciones de computadoras. Este servicio está dirigido tanto al colegio como al alumnado y a los profesores.

Nuestra iniciativa surge del deseo de aplicar y compartir los conocimientos adquiridos durante nuestra formación, ofreciendo una contribución significativa a nuestra comunidad educativa. A través de esta plataforma, buscamos facilitar el mantenimiento y la mejora de los equipos informáticos, esenciales para el proceso de enseñanza y aprendizaje.

Nuestro compromiso es proporcionar estos servicios de manera completamente gratuita, como una forma de retribuir y agradecer la educación que hemos recibido. Creemos firmemente en el poder de la colaboración y el apoyo mutuo, y estamos entusiasmados por la oportunidad de ayudar a mejorar las condiciones tecnológicas del colegio.

Esperamos que este esfuerzo conjunto beneficie a todos los miembros de nuestra escuela y ayude con un entorno de aprendizaje más eficiente y accesible.

Situación Problemática

En el contexto actual, nos enfrentamos a un desafío significativo relacionado con la operatividad de las computadoras (PC) en nuestro colegio. Observamos que existe un número considerable de equipos que, a pesar de su importancia para las actividades educativas, presentan problemas técnicos y no están funcionando correctamente.

Objetivo

Nuestro principal objetivo es garantizar el funcionamiento correcto y sostenible de las computadoras, asegurando su pleno uso y mantenimiento a lo largo del tiempo en el colegio. Reconocemos la importancia de una infraestructura tecnológica confiable y eficiente para apoyar las actividades educativas.

Nos guiamos por la estrategia de establecer objetivos alcanzables, enfocándonos en metas a corto plazo que sean fácilmente alcanzables. Esta metodología no solo promueve un progreso continuo, sino que también motiva y alienta a todos los miembros del equipo involucrados en el proyecto.

Los objetivos específicos se detallarán de manera concreta en la documentación correspondiente.

**Documentación del código frontend:**

Arreglo.html

Hemos implementado una opción especializada en nuestro sistema de soporte técnico que permite a nuestros técnicos seleccionar y gestionar las computadoras disponibles en el taller para su reparación. Esta funcionalidad está diseñada para optimizar el proceso de diagnóstico y reparación, debido a que cada tecnico puede variar su eleccion segun la cantidad de conocimiento que tenga, asegurando que cada equipo reciba la atención necesaria de manera eficiente.

Asesoramiento

Se utilizaria para garantizar una atención óptima y resolver cualquier inquietud que pueda surgir, hemos implementado un sistema de comunicación directa respecto a dudas,sugerencias e inquietudes. Este canal está diseñado para que usted pueda expresar sus dudas de manera clara y efectiva, permitiéndonos ofrecerle un diagnóstico preciso y oportuno.

Nuestro equipo de servicio técnico está a su disposición para proporcionarle la información y asistencia necesarias, asegurando así que todas sus consultas sean atendidas de forma profesional y eficiente.

Limpieza.html

Esta sección está diseñada específicamente para facilitar la solicitud de servicios de limpieza. Aquí encontrará toda la información necesaria para coordinar y gestionar sus necesidades de limpieza de manera eficiente.

Style.css

Esta sección está dedicada a establecer las características superficiales de la página, abarcando cada vista de manera individual. Proporciona una guía detallada sobre la apariencia del sitio, asegurando una presentación coherente y atractiva. Aquí se encuentran los elementos clave que conforman la estética y la estructura de cada vista.

Index.html

La vista principal de la página actúa como el núcleo central de organización y distribución de todas las demás vistas. Es la principal fuente de datos y estructura de la página, proporcionando la base sobre la cual se construyen y desarrollan las fases posteriores del sitio. Esta vista integra y unifica todos los componentes, asegurando una coherencia y funcionalidad óptimas en la experiencia del usuario.

Contactos.html

Esta sección está destinada a facilitar la comunicación directa con el personal del proyecto. Aquí se encuentran datos específicos como dirección, correo electrónico, teléfono, entre otros. Este recurso es utilizado para establecer contacto fuera del ámbito de asesoramiento de la página, permitiendo una comunicación más personalizada y detallada con nuestro equipo.

**Documentación de la Integración de Node.js en el Proyecto FixApp**

**¿Qué es Node.js?**

Node.js es un entorno de ejecución de JavaScript que se construye sobre el motor de JavaScript V8 de Google Chrome. A diferencia de los entornos de ejecución tradicionales de JavaScript que funcionan en el navegador, Node.js permite ejecutar código JavaScript en el servidor. Esto posibilita el desarrollo de aplicaciones del lado del servidor con JavaScript, un lenguaje que tradicionalmente se ha utilizado principalmente en el desarrollo del lado del cliente.

**Fundamentos y Ventajas de Node.js**

1. **Asincronía y No Bloqueo**: Node.js está diseñado para ser asincrónico y no bloqueante, lo cual significa que puede manejar múltiples solicitudes simultáneamente sin bloquear el hilo principal. Esto lo hace extremadamente eficiente en términos de rendimiento y escalabilidad, especialmente en aplicaciones de alta concurrencia como las aplicaciones web en tiempo real.

1. **Uso de JavaScript en el Backend**: Utilizar JavaScript tanto en el frontend como en el backend permite a los desarrolladores trabajar con un único lenguaje en toda la pila de la aplicación. Esto simplifica el desarrollo y facilita la colaboración entre equipos, ya que no se requiere aprender un nuevo lenguaje para el desarrollo del lado del servidor.

1. **Ecosistema de Paquetes (npm)**: Node.js cuenta con npm (Node Package Manager), uno de los ecosistemas de paquetes más grandes del mundo. Esto proporciona acceso a una vasta cantidad de librerías y módulos que pueden acelerar el desarrollo y reducir el tiempo necesario para implementar funcionalidades complejas.

1. **Alta Velocidad y Eficiencia**: Gracias al motor V8 de Google, Node.js ejecuta código JavaScript a alta velocidad. Además, la naturaleza no bloqueante de Node.js asegura que los recursos del servidor se utilicen de manera eficiente.

1. **Comunidad y Soporte**: Node.js tiene una comunidad activa y en crecimiento, lo cual significa que hay un amplio soporte disponible, desde documentación hasta foros de discusión y contribuciones de código abierto.

**Reorganización de la Estructura del Proyecto**

Para mejorar la visualización y evaluación de las distintas partes del proyecto, se ha decidido organizar las carpetas del proyecto FixApp en dos partes principales: frontend y backend.

**Justificación de la Reorganización**

La separación del proyecto en frontend y backend tiene varias ventajas:

1. **Claridad y Mantenimiento**: Una estructura organizada facilita la identificación y localización de archivos y directorios, lo cual mejora la mantenibilidad del proyecto.

2. **Colaboración**: Equipos diferentes pueden trabajar simultáneamente en el frontend y el backend sin interferencias. Esto es crucial en proyectos donde el desarrollo paralelo es necesario.

3. **Escalabilidad**: La división clara permite escalar cada parte de manera independiente. Por ejemplo, se pueden implementar servicios de backend en múltiples servidores mientras que el frontend puede ser servido desde un CDN.

4. **Pruebas y Despliegue**: Facilita la realización de pruebas específicas para cada parte y simplifica el proceso de despliegue, permitiendo desplegar cambios en el frontend o backend de manera independiente.

1. **Mejora del Rendimiento**: La separación permite optimizar cada parte de la aplicación según sus necesidades específicas. Por ejemplo, el backend puede ser optimizado para manejar solicitudes concurrentes mientras que el frontend puede ser optimizado para una carga rápida y eficiente en el navegador.

**Explicación del código de conexión a la base de datos:**

**Instalar dependencias:**

* mysql2: Permite que Node.js hable con MySQL.
* dotenv: Mantiene las contraseñas y configuraciones fuera del código en un archivo .env.
* Configurar variables de entorno:

**Crea un archivo .env en tu proyecto**.

Dentro del archivo .env, define:

* DB\_HOST: Dirección del servidor de la base de datos.
* DB\_USER: Nombre de usuario de la base de datos.
* DB\_PASSWORD: Contraseña de la base de datos.
* DB\_DATABASE: Nombre de la base de datos.

**Crear el archivo de conexión a la base de datos:**

**Crea un archivo db.js.**

Importa mysql2 y dotenv.

* Usa dotenv para cargar las variables de entorno.
* Configura la conexión a la base de datos con las variables del archivo .env.
* Conéctate a la base de datos y muestra un mensaje si la conexión es exitosa o hay un error.
* Exporta la conexión para usarla en otros archivos. Usar la conexión en tu aplicación: