**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 utilizaría 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.

**Justificación de la Integración de Bootstrap en el Proyecto FixApp**

Bootstrap ofrece diversas ventajas para el desarrollo rápido y prototipado de interfaces de usuario. Con su conjunto de componentes prediseñados y estilos CSS, es posible construir interfaces atractivas y funcionales de manera eficiente, ideal para proyectos con restricciones de tiempo.

Además, su enfoque "mobile-first" garantiza que las aplicaciones sean responsivas desde dispositivos móviles hasta escritorios, asegurando una experiencia consistente. Bootstrap también facilita mantener una apariencia visual uniforme mediante estilos predefinidos y componentes estructurados, mejorando la coherencia estética y la experiencia del usuario. Con un soporte activo y una comunidad robusta, ofrece recursos extensos de documentación y ejemplos para resolver problemas y personalizar proyectos según las necesidades específicas, combinando la facilidad de uso con la flexibilidad de la personalización.

**Importancia para la integridad y estilización de la página**

* **Integridad estructural:** Bootstrap proporciona una estructura coherente y bien definida para el desarrollo front-end, lo que facilita la organización y mantenimiento del código. Esto es esencial para proyectos grandes o equipos donde múltiples desarrolladores trabajan en el mismo código base.
* **Estilización coherente:** Al usar Bootstrap, te aseguras de que los elementos de tu página estén estilizados de manera uniforme, siguiendo las mejores prácticas de diseño web. Esto no solo mejora la apariencia visual, sino que también contribuye a una experiencia de usuario más agradable y profesional.

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

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.

**Node.js ofrece varias ventajas fundamentales que lo hacen ideal para el desarrollo de aplicaciones modernas:**

Node.js está diseñado con un enfoque en la asincronía y el no bloqueo, permitiendo manejar múltiples solicitudes simultáneamente sin bloquear el hilo principal. Esto optimiza el rendimiento y la escalabilidad, especialmente en aplicaciones web en tiempo real y de alta concurrencia.

El uso de JavaScript tanto en el frontend como en el backend simplifica el desarrollo al permitir a los desarrolladores trabajar con un único lenguaje en toda la pila de la aplicación. Esto facilita la colaboración entre equipos y elimina la necesidad de aprender un nuevo lenguaje para el desarrollo del lado del servidor.

Node.js cuenta con npm (Node Package Manager), uno de los ecosistemas de paquetes más extensos del mundo. Con acceso a una amplia variedad de librerías y módulos, facilita la implementación de funcionalidades complejas y acelera el desarrollo de aplicaciones.

Gracias al motor V8 de Google, Node.js ejecuta código JavaScript a alta velocidad, optimizando la eficiencia del servidor. Su naturaleza no bloqueante asegura una utilización eficiente de los recursos del sistema.

Node.js cuenta con una comunidad activa y en expansión, lo cual garantiza un amplio soporte a través de documentación, foros de discusión y contribuciones de código abierto, asegurando que los desarrolladores siempre encuentren ayuda y recursos para mejorar sus proyectos.

**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

Se realizo la conexión con el archivo server.js

Este mismo implementa la conexión a la base de datos con las siguientes herramientas y librerías:

const express = require('express'):

* Express permite crear servidores web, manejar rutas, y gestionar solicitudes y respuestas HTTP de manera eficiente.

const mysql = require('mysql2'):

* La biblioteca mysql2 es una librería de Node.js que permite interactuar con bases de datos MySQL y MariaDB desde una aplicación Node.js.

const path = require('path'):

* Facilita la manipulación y resolución de rutas en sistemas de archivos, permitiendo trabajar de manera consistente en diferentes sistemas operativos (Windows, macOS, Linux).

const bodyParser = require('body-parser'):

* La biblioteca body-parser es un middleware para aplicaciones Express en Node.js que se utiliza para analizar y procesar los cuerpos de las solicitudes HTTP.

Documentacion del Codigo del archivo server.js:

const connection = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '',

database: 'appfix'

});

Este fragmento de código permite conectarse de forma local a la base de datos utilizando el método mysql.createConnection de la librería mysql2.

connection.connect((err) => {

if (err) {

console.error('Error conectando a la base de datos:', err);

return;

}

console.log('Conexión a la base de datos establecida.');

});

Con esta fracción de código previene cualquier tipo de error de conexión a la base de datos, si ocurriese algún tipo de error se le será notificado con un mensaje en la consola.

Funciona como un try y catch.

Las siguientes líneas de código son de enrutamiento y dirección del archivo contacto.html

app.use(express.static(path.join(\_\_dirname, 'public')));

app.use(bodyParser.urlencoded({ extended: false }));

app.get('/', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'contacto.html'));

});

Luego de ser enrutada lo que se recogio del formulario, es redirigida la información hacia la base de datos por medio de un método connection de la propia librería de mysql2.

app.post('/contact', (req, res) => {

const { name, email, phone, message } = req.body;

const query = 'INSERT INTO contact\_requests (name, email, phone, message) VALUES (?, ?, ?, ?)';

Se hace una consulta a la base de datos y si no resulta exitosa se enviara un mensaje notificando dicho error, caso contrario la consulta será exitosa; por ende se envía un mensaje que haga mención a ello.

connection.query(query, [name, email, phone, message], (err, results) => {

if (err) {

console.error('Error realizando la consulta:', err);

return res.status(500).send('Error al enviar la consulta.');

}

res.send('Consulta enviada correctamente.');

});

});