TECNOLOGICO NACIONAL DE MEXICO

CAMPUS ORIZABA

INGENIERIA EN SISTEMAS COMPUTACIONALES

FRAMEWORKS PARA APLICACIONES ENRIQUECIDAS DE INTERNET

GRUPO 7gE

13-14 HRS

CATEDRATICO

MONICA RUIZ MARTINEZ

EQUIPO

- **Bermúdez López Roberto Sebastián**
- **Hernández Gómez Richard de Jesús**
- **Muñoz Hernández Vania Lizeth**

UNIDAD: 2

Título de proyecto:

"BUZON DE QUEJAS DEL INSTITUTO TECNOLOGICO NACIONAL DE ORIZABA"

Nombre de la página:

SISTEMABUHO DE GESTION DE QUEJJAS

FECHA DE ENTREGA 31 DE MARZO DEL 2025

Introducción

El presente informe tiene como objetivo detallar el proceso de desarrollo de la página "Sistema Búho de Gestión de Quejas" implementada en Vue.js. Esta aplicación permite a los estudiantes del Instituto Tecnológico de Orizaba presentar quejas y sugerencias de manera anónima, mejorando la accesibilidad y eficiencia del proceso. En este documento se describirá el proceso de desarrollo, los desafíos enfrentados, las estrategias de resolución de problemas y una evaluación general de Vue.js como herramienta para el desarrollo de aplicaciones web enriquecidas.

El propósito principal de esta aplicación es mejorar el proceso de recepción y gestión de quejas, brindando una plataforma accesible para que los estudiantes expresen sus inquietudes de manera rápida y sencilla. También cuenta con herramientas de seguimiento y generación de estadísticas, lo que permite analizar la información recopilada y tomar decisiones basadas en datos.

El desarrollo de esta aplicación se llevó a cabo siguiendo una metodología estructurada, utilizando herramientas y bibliotecas como Vue Router para la navegación, Vuex para la gestión del estado y MySQL para el almacenamiento de datos. A lo largo del proceso, se presentaron diversos desafíos técnicos y de diseño, los cuales fueron abordados con estrategias específicas para optimizar el rendimiento y la experiencia del usuario.

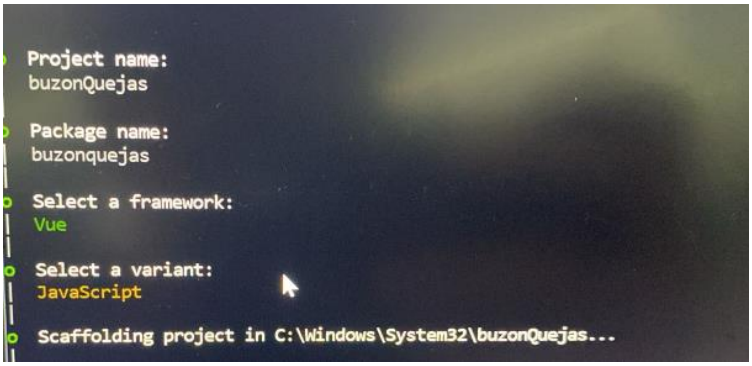
En este informe, se detallará el proceso de desarrollo de la aplicación paso a paso, los desafíos enfrentados durante su implementación, las estrategias utilizadas para resolver estos problemas y una evaluación general sobre la experiencia de trabajar con Vue.

Proceso de Desarrollo

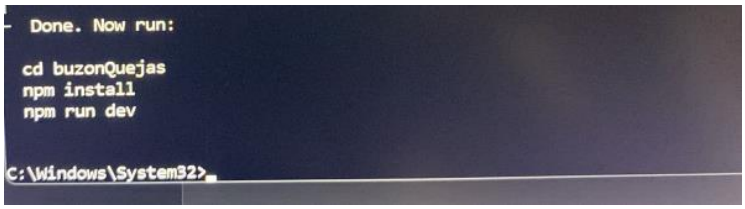
El desarrollo del " Sistema Búho de Gestión de Quejas" utilizando Vue.js se llevó a cabo siguiendo un proceso estructurado, asegurando una implementación ordenada y eficiente.

- En primer lugar, se creó el proyecto utilizando Vue.js con la variante de JavaScript. Para ello, se ejecutó el siguiente comando en la terminal:

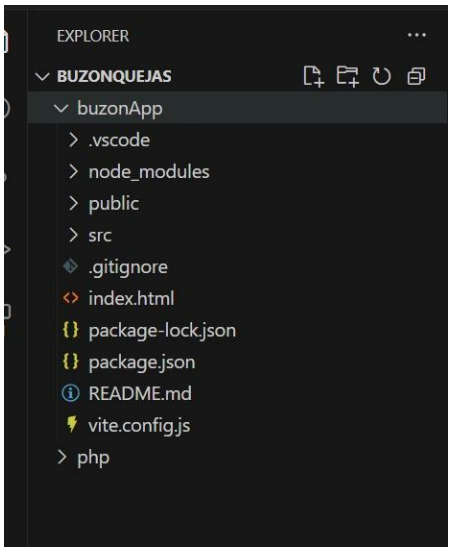
npm create vue@latest buzonQuejas



Durante la configuración del proyecto, se seleccionó Vue como framework y JavaScript como variante principal, en lugar de TypeScript. Una vez finalizado este proceso, se ingresó al directorio del proyecto y se instalaron las dependencias necesarias con los siguientes comandos:



Esto permitió iniciar el entorno de desarrollo y verificar que la estructura inicial del proyecto estaba correctamente configurada. Después de ello se movieron los archivos a las 2 carpetas principales que son las siguientes:



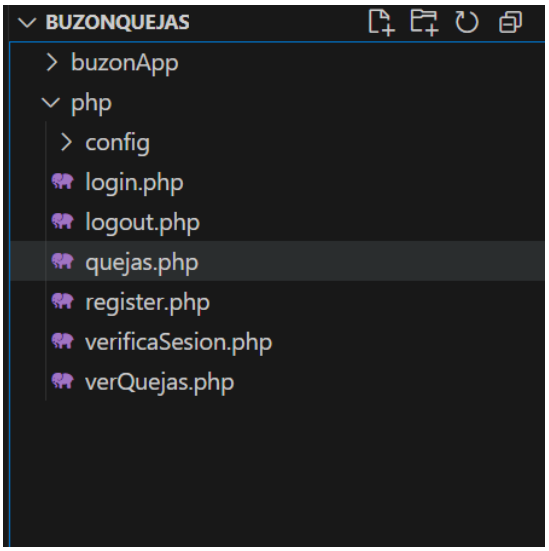
En el archivo router/index.js se configuro Vue Router, que es el encargado de gestionar la navegación entre las diferentes páginas de la aplicación sin necesidad de recargar la página el cual quedo de la siguiente manera.

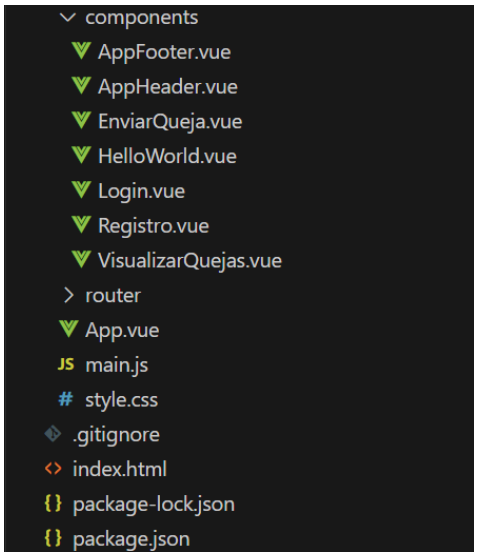
buzonApp > src > router > JS index.js > routes

```
1   import { createRouter, createWebHistory } from 'vue-router';
2
3   const routes = [
4     { path: '/', component: () => import('../components/HelloWorld.vue') },
5     {
6       path: '/registro',
7       component: () => import('../components/Registro.vue')
8     },
9     {
10      path: '/login',
11      component: () => import('../components/Login.vue')
12    },
13    {
14      path: '/enviar-queja',
15      component: () => import('../components/EnviarQueja.vue'),
16      meta: { requiresAuth: true }
17    },
18    {
19      path: '/visualizar-quejas',
20      component: () => import('../components/VisualizarQuejas.vue'),
21      meta: { requiresAuth: true, requiresAdmin: true }
22    }
```

Para la implementación del proyecto, organizamos los archivos en una estructura adecuada. Creamos la carpeta php, donde almacenamos los archivos del backend encargados de gestionar la autenticación, el registro de usuarios y la administración de quejas.

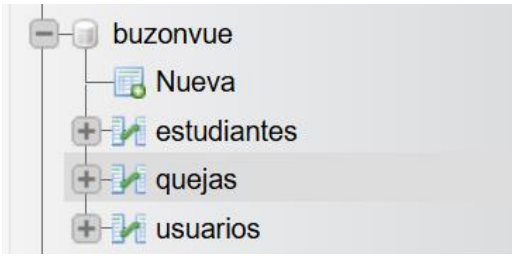
Por otro lado, en la carpeta components, dentro del proyecto en Vue, desarrollamos los distintos componentes de la interfaz, como los formularios de inicio de sesión, registro y envío de quejas.





Además de la estructura de archivos, creamos la base de datos buzonvue, que es fundamental para almacenar y gestionar la información del sistema. En ella definimos tres tablas principales:

- **estudiantes**: Contiene la información de los alumnos registrados en el sistema, como su número de control y carrera.
- **quejas**: Registra las quejas enviadas por los estudiantes, incluyendo el mensaje, el estado de la queja (pendiente, en proceso, resuelta o rechazada) y la relación con el estudiante que la envió.
- **usuarios**: Almacena los datos de autenticación, permitiendo la gestión del acceso al sistema que contiene los datos de numero de control, nombre completo, contraseña y tipo de usuario (estudiante o administrador).



Esta estructura permite organizar la información de manera eficiente, facilitando la autenticación de los estudiantes y el seguimiento de las quejas presentadas.

Dentro de nuestro proyecto, los marcos utilizados usados fueron los siguientes:

- **Vue**.js: Framework de JavaScript donde desarrollamos el frontend.
- **PHP**: Dentro de este apartado creamos el backend para gestionar las peticiones y la conexión con la base de datos.
- **MySQL**: Gestionamos las bases de datos donde se almacenan las quejas y la información de los estudiantes que se registran dentro de la página.
- **XAMPP**: Permitió la ejecución de Apache, PHP y MySQL en local.

Dentro de nuestro proyecto, las bibliotecas de Vue fueron fundamentales ya que nos facilita la creación de interfaces interactivas y dinámicas en la web, en esta parte del codigo v-model permite enlazar los campos del formulario con las variables del estado que en este caso es con la contraseña.

```
27              id="contra"
28              v-model="contra"
29              maxlength="20"
30              required
31              title="Máximo 20 caracteres"
32            >
33          </div>
34
35          <p v-if="mensaje" class="mensaje">{{ mensaje }}</p>
```

Dentro de las bibliotecas, además del v-model, nos permiten que la aplicación Vue sea interactiva, organizada y eficiente, facilitando la navegación, la comunicación con el backend y la administración de estilos. Algunas otras fueron:

1. **Directivas de Vue:**
   - **v-model**
   - **v-on o @**
   - **v-if / v-show**

2. **Vue Router:**
   - **<router-link>**
   - **useRouter()**

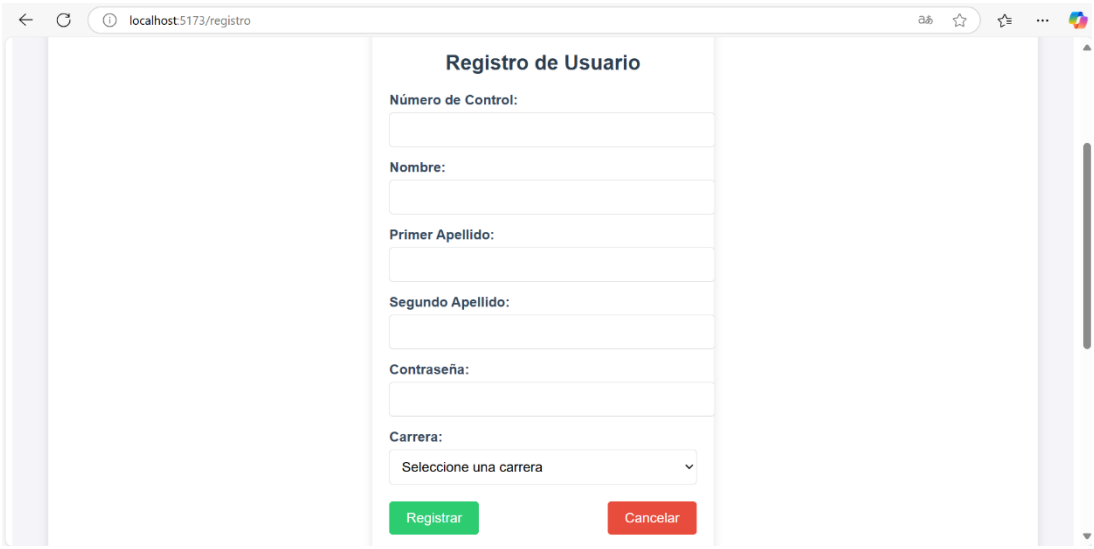
3. **Composition API:**
   - **ref()**
   - **defineComponent**

Con la estructura de archivos bien organizada la página ya es completamente accesible y funcional.

Ahora, la aplicación se ejecuta sin problemas, permitiendo que los estudiantes inicien sesión, envíen sus quejas y realicen un seguimiento de su estado. Gracias a la conexión establecida entre Vue.js y el backend en PHP, las peticiones se procesan correctamente y la información se almacena y recupera de la base de datos de manera eficiente.

Alumnos:

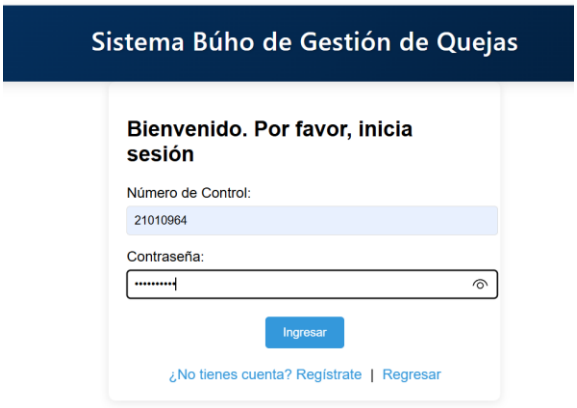
Completamos el registro del usuario (el registro solamente sirve para agregar estudiantes, por cuestiones de seguridad los administradores se agregan directamente en script SQL en la base de datos) para que podamos tener acceso a el buzón de quejas, de lo contrario tendremos el acceso denegado por no poseer una cuenta.



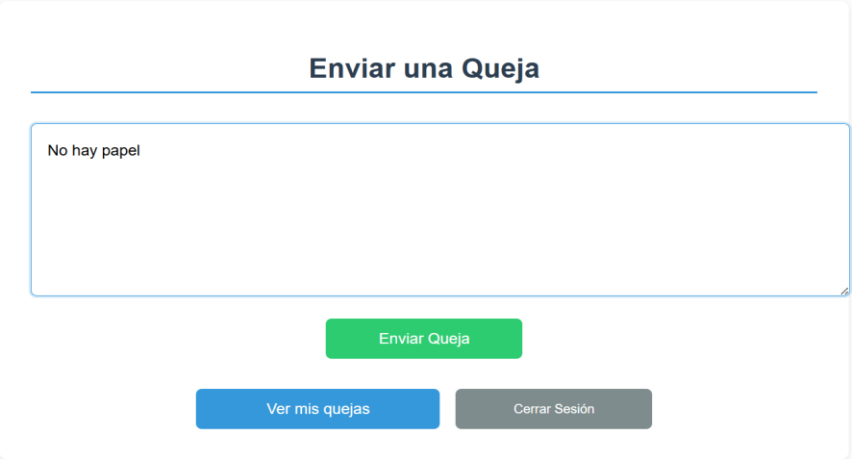
Verificamos que en nuestra base de datos este almacenando los registros de los usuarios registrados.

id	nombre	primApe	segApe	contra	tipo
12345678	Admin01	AdminApellido1	AdminApellido2	$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2...	admin
21010964	Richard	Hernandez	Gomez	$2y$10$..SL1T6AbOWDuJH7Ty3LBO.vsJa017O8v22R4Gz8tpk...	estudiante

En este caso el usuario registrado fue 21010964, que vemos que ya es registrado correctamente, para después pueda iniciar sesión y registrar su queja, para ello, el usuario inicia sesión para verificar que ha cumplió correctamente su registro.



Después de iniciar sesión, los estudiantes pueden escribir sus inquietudes o sugerencias a través de un apartado que les permite redactar su queja, enviarla y asegurarse de que será almacenada correctamente en la base de datos para su posterior revisión y gestión por parte de la administración.



En este caso notamos que el usuario registró dos quejas de la cual una de ellas esta siendo procesada y otra pendiente, de igual forma en nuestra base de datos las quejas se registran correctamente.

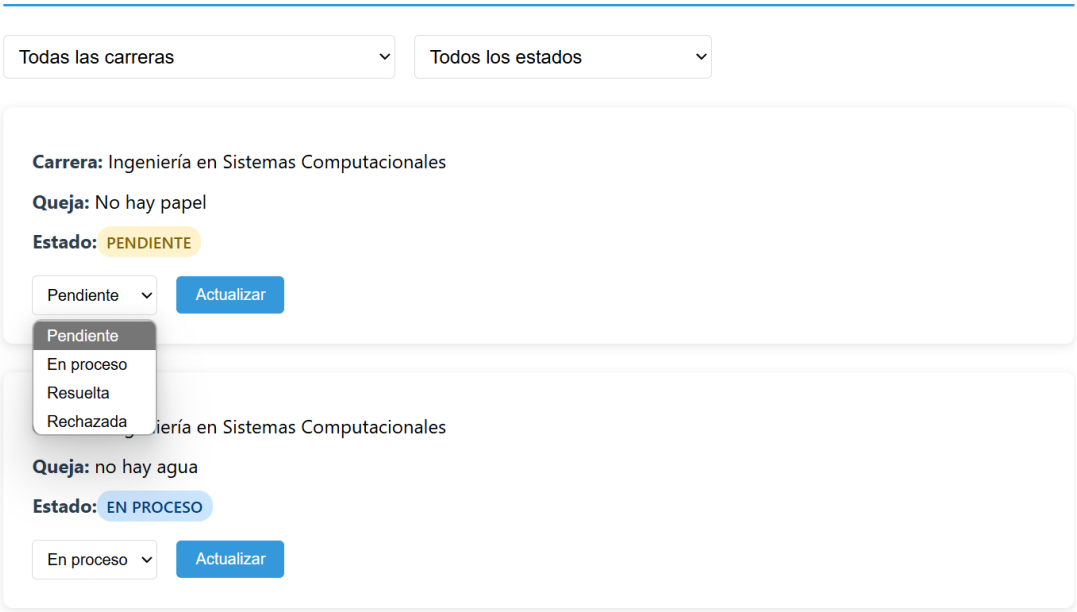
id	mensaje	id_estudiante	carrera	estado
1	No hay papel	21010964	Ingeniería en Sistemas Computacionales	pendiente
2	no hay agua	21010964	Ingeniería en Sistemas Computacionales	en_proceso

Las quejas se registran con éxito en nuestra base de datos. Notamos que algunas se encuentran en estado "pendiente" y otra se está llevando a cabo en "en proceso", como usuario puede revisar el estatus de su queja. Por defecto, cuando un usuario registra una queja, esta queda en estado con el ENUM "pendiente", permitiendo que posteriormente el administrador pueda revisarla y actualizar su estado conforme se le dé seguimiento y solución a la queja.

Administrador:

Inicia sesión con cuenta de administrador para tener acceso a la visualización de las quejas de los alumnos.

El usuario con rol de administrador tiene la autoridad para gestionar las quejas, modificando su estado a "en proceso", "atendida", “resuelta” o "rechazada", según corresponda.



Desafíos Enfrentados

Problema 1:

Inicialmente, al intentar realizar solicitudes a los archivos PHP ubicados en buzonQuejas/php, el navegador bloqueaba las peticiones, lo que impedía la comunicación adecuada entre la aplicación en Vue.js y el backend en PHP alojado en XAMPP.

Dentro de consola nos mostraba la siguiente línea de texto:

> *Access to XMLHttpRequest at 'http://localhost/buzonQuejas/php/endpoint.php' from origin 'http://localhost:5173' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.*

Este error indica que la aplicación Vue.js (corriendo en http://localhost:5173) intenta hacer una solicitud a un archivo PHP en el servidor XAMPP (http://localhost/buzonQuejas/php/endpoint.php), pero debido a que no se ha configurado adecuadamente el CORS en el servidor, el navegador bloquea la solicitud.

Problema 2:

El problema ocurrió porque el script en Vue enviaba datos en formato JSON al servidor PHP, pero PHP no estaba configurado para recibir correctamente el contenido. Esto causó que

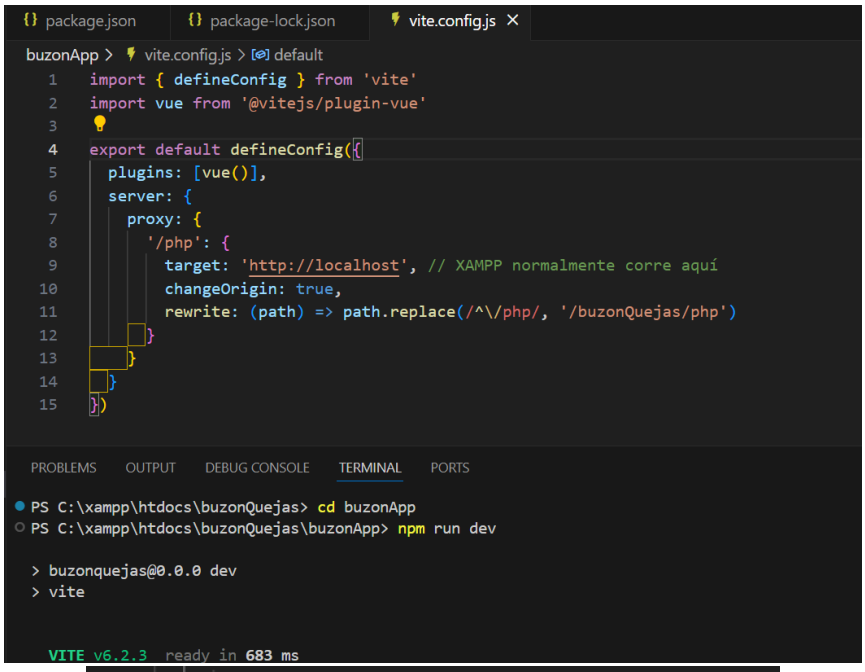
file_get_contents("php://input"

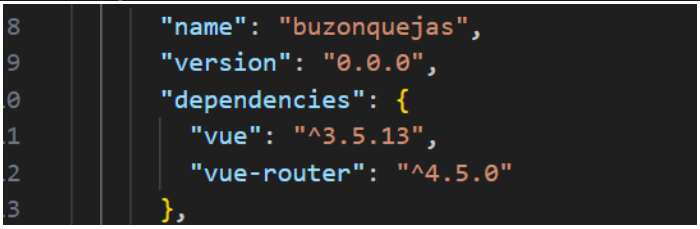
devolviera un valor vacío o null, impidiendo que la queja del usuario se almacenara correctamente en la base de datos. Además, PHP no reconocía el JSON porque no se especificaron correctamente los encabezados HTTP necesarios para interpretar el contenido y permitir el acceso desde diferentes orígenes.

Estrategias de Resolución de Problemas

Solución 1: Dentro del código, nos enfrentamos a la problemática del bloqueo de las peticiones debido a CORS. Para resolverlo, configuramos adecuadamente el proxy en el archivo vite.config.js, lo que permitió que la aplicación Vue.js pudiera comunicarse correctamente con el backend en PHP alojado en XAMPP.

De esta manera, el navegador ya no bloqueó las solicitudes, ya que la redirección se maneja de forma interna por el servidor de desarrollo de Vite, actuando como intermediario entre el frontend y el backend.



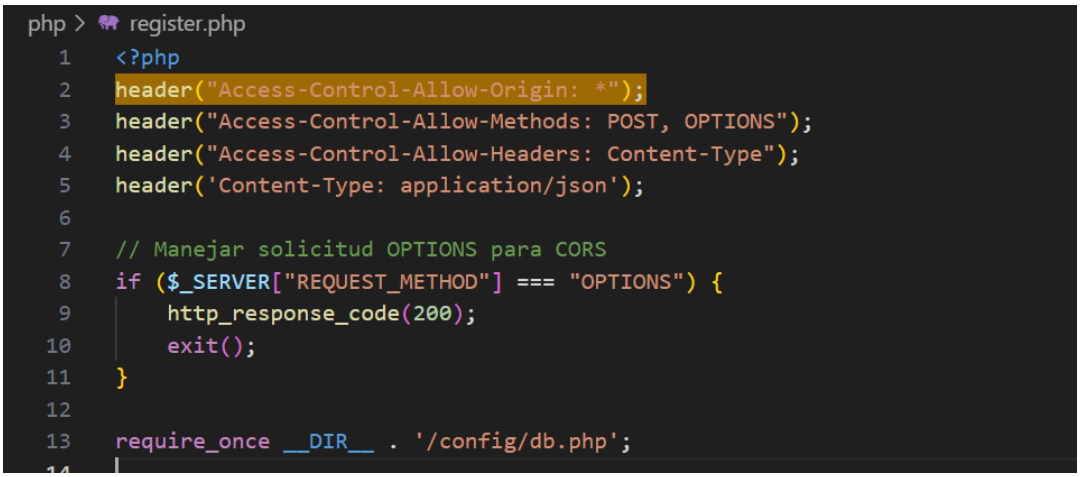


Fue necesario modificar y configurar correctamente las dependencias dentro del proyecto, porque nos permitió crear rutas dinámicas sin necesidad de recargar la página, donde los usuarios deben moverse entre distintas secciones (registro, login, enviar quejas y visualizar quejas).

Solución 2:

Para resolver el problema, se implementaron las siguientes mejoras en el archivo PHP:

1. Se configuraron correctamente las cabeceras HTTP para permitir el acceso desde cualquier origen y aceptar contenido JSON.
2. Se utilizó file_get_contents("php://input") para obtener el cuerpo de la solicitud y json_decode() para convertirlo en un array de PHP.
3. Se verificó que los datos llegaran correctamente al servidor y se envió una respuesta JSON de confirmación.



Con esta solución, el servidor PHP ahora puede recibir y procesar datos en formato JSON sin problemas, nos aseguramos que la comunicación entre Vue y PHP sea fluida.

Conclusiones

El desarrollo del Sistema Búho de Gestión de Quejas nos permitió aprender a resolver problemas técnicos importantes. Uno de los principales obstáculos fue que los datos enviados desde el frontend en Vue.js no llegaban correctamente al backend en PHP, lo que dificultaba almacenar las quejas en la base de datos. Para solucionarlo, tuvimos que ajustar la configuración del servidor PHP para que aceptara solicitudes en formato JSON, y también tuvimos que cambiar la forma en que enviábamos los datos desde Vue, asegurándonos de que estuvieran bien estructurados.

Este proyecto nos enseñó mucho sobre cómo debe fluir la información entre las diferentes partes de una aplicación web, y lo importante que es tener la configuración correcta para que todo funcione bien. También aprendimos a depurar y solucionar errores de manera más efectiva. A través de este proceso, no solo mejoramos nuestras habilidades técnicas, sino que también nos dimos cuenta de lo valioso que es trabajar en equipo y tener una buena comunicación.

Referencias electrónicas

1. Mozilla Developer Network (MDN). (s.f.). Using Fetch. Mozilla. Recuperado el 31 de marzo de 2025, de https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
2. PHP Manual. (s.f.). json_decode - Manual de PHP. PHP.net. Recuperado el 31 de marzo de 2025, de https://www.php.net/manual/es/function.json-decode.php
3. Vue.js. (s.f.). Vue Router: The official router for Vue.js. Vue.js. Recuperado el 31 de marzo de 2025, de https://router.vuejs.org/
4. MySQL. (s.f.). MySQL 8.0 Reference Manual. MySQL. Recuperado el 31 de marzo de 2025, de https://dev.mysql.com/doc/refman/8.0/en/
5. W3C. (s.f.). Cross-Origin Resource Sharing (CORS). W3C. Recuperado el 31 de marzo de 2025, de https://www.w3.org/TR/cors/