

Prueba Técnica - Implementación de un Chat en Tiempo Real con Laravel

Objetivo

Desarrollar una aplicación de chat en tiempo real utilizando Laravel. La aplicación debe permitir a los usuarios autenticados enviar y recibir mensajes de forma instantánea, utilizando **sockets, eventos y colas de Laravel**.

Requisitos Técnicos

Backend - Laravel

1. **Autenticación de Usuarios**
 - Implementar un sistema de autenticación con Laravel (usando Laravel Breeze, Jetstream o Sanctum).
 - Solo los usuarios autenticados pueden enviar y recibir mensajes.
 2. **Mensajería en Tiempo Real**
 - Implementar un sistema de chat en tiempo real usando **Laravel WebSockets** y eventos de Laravel.
 - Cuando un usuario envía un mensaje, este debe transmitirse automáticamente a los demás usuarios conectados.
 3. **Colas de Trabajo (Queues)**
 - Los mensajes deben enviarse a una **cola de trabajo** antes de ser guardados en la base de datos.
 - Utilizar el sistema de colas de Laravel para procesar los mensajes en segundo plano.
 4. **Eventos de Laravel**
 - Implementar eventos y listeners para gestionar la lógica de mensajería.
 - Emitir un evento cuando un usuario envía un mensaje y capturarlo en el frontend mediante **WebSockets**.
-

Frontend - (Libre Elección)

El candidato puede elegir la tecnología frontend que prefiera (React, Vue, Angular, Livewire, etc.), pero debe cumplir con:

- Un formulario para enviar mensajes.
- Un área donde se listen los mensajes en tiempo real.
- Mostrar el nombre del usuario junto con cada mensaje.

Si usa **Livewire** en Laravel, debe manejar la actualización del chat sin necesidad de recargar la página.

Requisitos Adicionales

- Usar **MySQL** o **PostgreSQL** como base de datos.
 - Subir el código a un repositorio en **GitHub** o **GitLab** con un README detallado explicando la configuración del entorno.
 - Documentar los pasos para ejecutar el proyecto.
 - Usar buenas prácticas en el código y seguir el estándar **PSR-4** de PHP.
-

Evaluación

Se evaluará lo siguiente:

- ✓ Correcta implementación de **Laravel WebSockets**.
 - ✓ Uso de **eventos y listeners** para manejar los mensajes.
 - ✓ Uso de **colas para procesamiento asíncrono**.
 - ✓ Diseño estructurado del código y buenas prácticas.
 - ✓ Funcionalidad completa en el **frontend** (actualización en tiempo real sin recargar la página).
-

Extras (Opcionales, pero suman puntos)

- ✓ Permitir **salas de chat** entre múltiples usuarios.
- ✓ Agregar **notificaciones** cuando llega un nuevo mensaje.
- ✓ Implementar un sistema de **visto** para los mensajes.