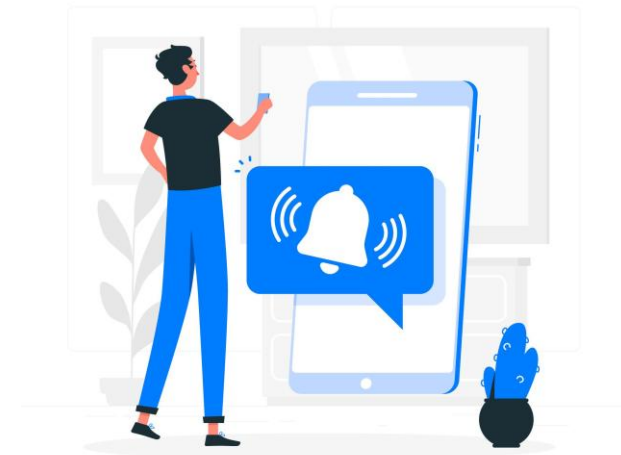


Conexiones en tiempo real (RTC)

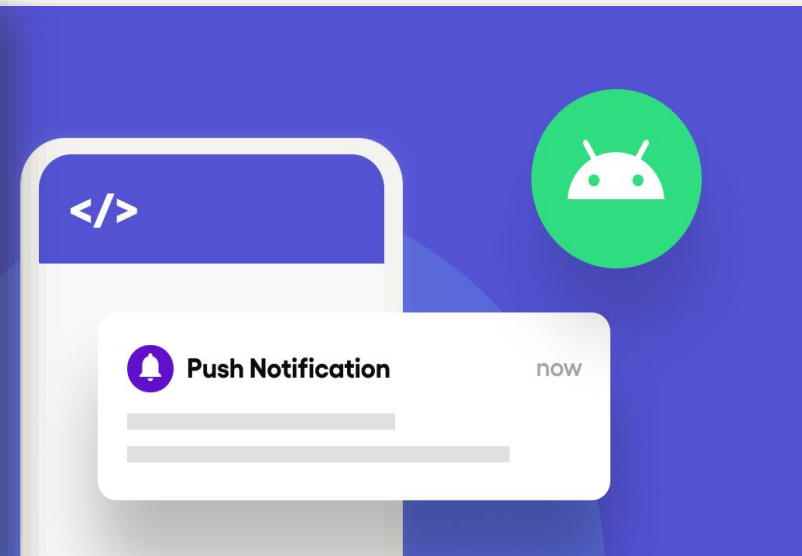
Presentado por: Darío Narváez

Licenciado en Informática
Universidad de Nariño
2025



Conexiones en tiempo real

Son métodos de comunicación que permiten el intercambio de datos sin latencia significativa. Se usan en chat, notificaciones, juegos en línea, etc.

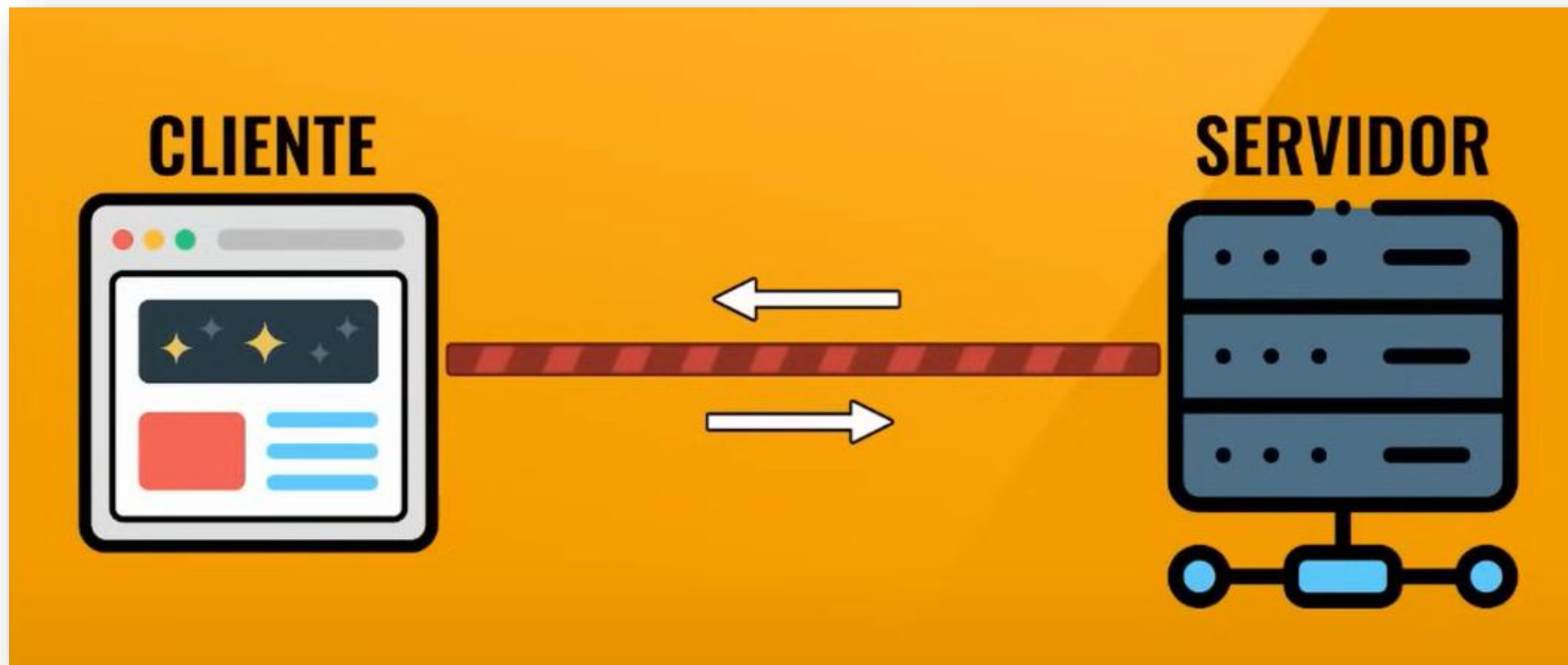


Que son los websockets?

- **WebSockets** es una tecnología avanzada que hace posible abrir una sesión de comunicación interactiva entre **el navegador del usuario y un servidor**. Con esta API, puede enviar mensajes a un servidor y recibir respuestas controladas por eventos sin tener que consultar al servidor para una respuesta. (Mozilla Developer)
- Funcionan sobre HTTP (handshake) y utilizan el protocolo WS por medio de una conexión TCP y el puerto 6001 (por defecto). Este hace posible establecer una conexión continua full-duplex, entre un cliente y servidor.
- Un cliente websocket podría ser el navegador del usuario, pero el protocolo es una plataforma independiente.



Que son los websockets



Comunicación abierta – Bidireccional



Que es el sdk de Pusher?

- El SDK de Pusher permite a los desarrolladores enviar y recibir mensajes en tiempo real.
- El SDK de Pusher permite a los desarrolladores integrarse con el entorno de nube en el idioma que prefieran.
- El SDK de Pusher permite a los desarrolladores configurar de una forma sencilla el envío de información a sus aplicaciones.



AJAX VS WEBSOCKETS



Conexión de una
sola vía

Hace una petición
cada vez

Se usa para solicitar
información

No sustituye a
Websockets



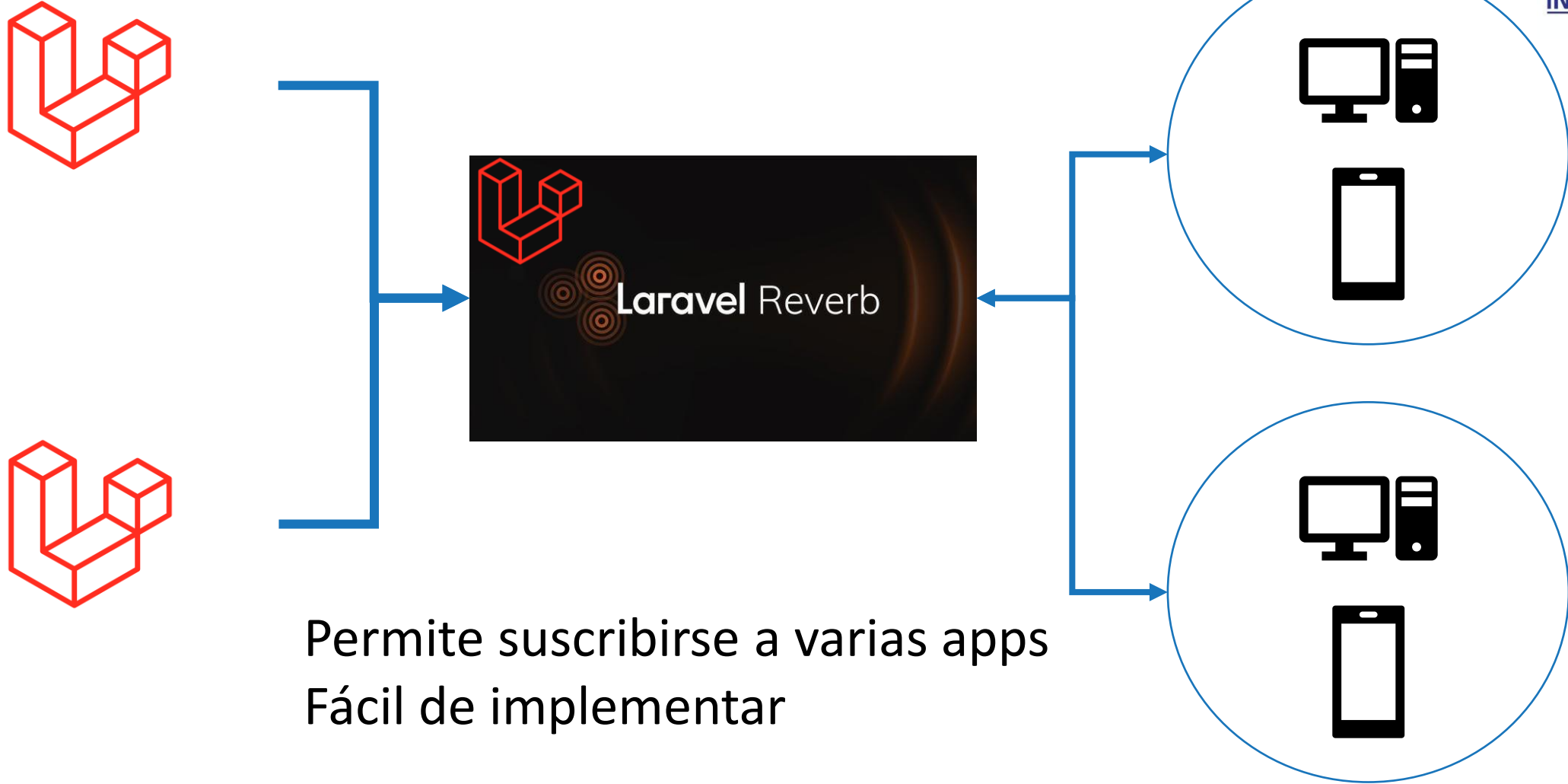
Conexión de
doble vía

Mantiene el canal
abierto

Comunicación en
tiempo real

No sustituye a
Ajax

Como funciona un servidor RTC



Configuración de un servidor con LARAVEL REVERB

Requisitos:

- Php 8.2
- Aplicación LARAVEL versión 11 (o superiores)
- Node



1. INSTALACIÓN DE LARAVEL REVERB EN UN PROYECTO LARAVEL

```
PS C:\Users\Dario\Documents\Apps\Laravel\reverb-server>php artisan install:broadcasting
```

```
INFO Published 'broadcasting' configuration file.
```

```
INFO Published 'channels' route file.
```

```
Would you like to install Laravel Reverb? (yes/no) [yes]
```

```
> yes
```

```
Would you like to install and build the Node dependencies required for broadcasting? (yes/no) [yes]
```

```
> yes
```

php artisan install:broadcasting



2. CONFIGURACIÓN DE CLIENTE EN SERVIDOR.

```
[  
    'key' => 'uyibawir5vww4ampl0lm',  
    'secret' => 'ntxgfndwxndzsiyvrttd',  
    'app_id' => '359574',  
    'options' => [  
        'host' => 'localhost',  
        'port' => "8080",  
        'scheme' => env('REVERB_SCHEME_2', 'http'),  
        'useTLS' => env('REVERB_SCHEME_2', 'https') === 'https',  
    ],  
    'allowed_origins' => ['*'],  
    'ping_interval' => env('REVERB_APP_PING_INTERVAL_2', 60),  
    'activity_timeout' => env('REVERB_APP_ACTIVITY_TIMEOUT_2', 30),  
    'max_message_size' => env('REVERB_APP_MAX_MESSAGE_SIZE_2', 10_000),  
]
```

Modificar archivo: config/reverb.php



3. Poner en servicio el servidor

```
PS C:\Users\Dario\Documents\Apps\Laravel\reverb-server> php artisan reverb:start --host=127.0.0.1
```

Php artisan config:cache

Php artisan reverb:start --host=127.0.0.1



4. CONFIGURACIÓN DE CLIENTE.

- Instalación de LARAVEL echo y PUSHER

```
PS C:\Users\Dario\Documents\Apps\Laravel\reverb-chat> npm install --save laravel-echo pusher-js
```

npm install --save laravel-echo pusher-js

- Instalación de pusher

```
PS C:\Users\Dario\Documents\Apps\Laravel\reverb-chat> composer require pusher/pusher-php-server
```

Composer require pusher/pusher-php-server



5. CONFIGURACIÓN DE CLIENTE.

```
PS C:\Users\Dario\Documents\Apps\Laravel\reverb-chat>php artisan install:broadcasting
```

```
INFO Published 'broadcasting' configuration file.
```

```
INFO Published 'channels' route file.
```

```
Would you like to install Laravel Reverb? (yes/no) [yes]
```

```
> NO
```

```
Would you like to install and build the Node dependencies required for broadcasting? (yes/no) [yes]
```

```
> yes
```

php artisan install:broadcasting



6. CONFIGURACIÓN DE CLIENTE.

Configurar conexión a servidor LARAVEL REVERB

```
BROADCAST_CONNECTION=reverb
QUEUE_CONNECTION=sync

REVERB_APP_ID=359574
REVERB_APP_KEY=uyibawir5vwv4amp101m
REVERB_APP_SECRET=ntxgfndwxndzsiyvrttd
REVERB_HOST="laravel.reverb.test"
REVERB_PORT=8080
REVERB_SCHEME=http

VITE_REVERB_APP_KEY="${REVERB_APP_KEY}"
VITE_REVERB_HOST="${REVERB_HOST}"
VITE_REVERB_PORT="${REVERB_PORT}"
VITE_REVERB_SCHEME="${REVERB_SCHEME}"
```

Modificar archivo: .env

7. CONFIGURACIÓN DE CLIENTE.

Configurar conexión a servidor LARAVEL REVERB

```
import Pusher from 'pusher-js';
import Echo from 'laravel-echo';
window.Pusher = Pusher;
window.Echo = new Echo({
  broadcaster: 'reverb',
  key: import.meta.env.VITE_REVERB_APP_KEY,
  wsHost: import.meta.env.VITE_REVERB_HOST,
  wsPort: import.meta.env.VITE_REVERB_PORT ?? 80,
  wssPort: import.meta.env.VITE_REVERB_PORT ?? 443,
  forceTLS: (import.meta.env.VITE_REVERB_SCHEME ?? 'https') === 'https',
  enabledTransports: ['ws', 'wss'],
  authEndpoint: "/broadcasting/auth", // Asegura que Laravel autentica la suscripción
  auth: {
    headers: {
      "X-CSRF-TOKEN": document.querySelector('meta[name="csrf-token"]')
        .getAttribute('content')
    },
    withCredentials: true
  }
});
```

Modificar archivo: resources/js/echo.js


```
81     const chatBox = document.getElementById('chat-box');
82     window.Echo.private('chat-channel')
83         .listen('.Chat', (data) => {
84         ▶     var id = document.querySelector(
85             ▶     `#data_${data.message.id}`) // document.get
86         if (userId !== data.message.user.id) {
87         ▶     const chatBox = document.getElementById('ch
88         const dataM = {
89             id: 00,
90             user_name: data.message.user.name,
91             user_id: data.message.user.id,
92             user: data.message.user, // Simulación
93             message: data.message.message,
```

Archivo: resources/views/dashboard.blade.php



8. CONFIGURACIÓN DE CLIENTE.

- Compilar los assets

```
PS C:\Users\Dario\Documents\Apps\Laravel\reverb-chat> npm run build
```

npm run build

```
PS C:\Users\Dario\Documents\Apps\Laravel\reverb-chat> npm run dev
```

npm run dev



```
public function __construct(Message $message)
{
    $this->message = $message;
}
public function broadcastOn()
{
    Log::info('Message sent 1');
    return new PrivateChannel('chat-channel');
}
public function broadcastAs()
{
    return 'Chat';
}
```

Archivo: app/Events/MessageSent

```
Broadcast::channel('chat-channel', function ($user) {  
    Log::info('chat autenticado: ' . $user->id);  
    return $user; // 0 una condición para autorizar al usuario  
});
```

Archivo: app/Events/MessageSent

