

## Reto IA 6

Criterio	REST + Polling (cada 5s)	WebSocket	Server-Sent Events (SSE)	Mi evaluación
Complejidad en el cliente (1–5)	1 – fetch + setInterval, cero dramas	4 – manejo de conexión, eventos, reconexión	2 – EventSource, casi plug & play	Polling es el mas simple y sencillo, tanto WebSockets como SSE suele ser un poco mas llamativo para el cliente
Uso de recursos del servidor (1000 clientes)	<b>Alto</b> $1000 \times (12 \text{ req/min}) = 12\,000 \text{ req/min}$ aunque no cambie nada	<b>Medio</b> 1000 conexiones abiertas, pero solo envía cuando hay cambios	<b>Bajo-Medio</b> 1000 conexiones HTTP persistentes, solo push	Polling es de trabajo constante; WebSocket y SSE solo trabajan cuando hay cambios.
Latencia del cambio de inventario	<b>Hasta 5 s</b> (depende del intervalo)	<b>Muy baja</b> casi instantánea	<b>Muy baja</b> push inmediato	polling pierde por que suele ser el más lento. WebSocket y SSE suelen ser muy rápidas
Pérdida breve de conexión	<b>Tolerante</b> la siguiente petición “lo arregla”	<b>Más delicado</b> hay que reintentar y resincronizar estado	<b>Muy buena</b> reconexión automática + Last-Event-ID	SSE reconecta solo y continúa el stream; WebSocket y Polling requieren intervención
Facilidad con mock server	<b>Muy fácil</b> JSON Server, JSONPlaceholder, Express	<b>Media</b> necesitas soporte WS real	<b>Fácil</b> Express / NestJS lo soportan rápido	WebSocket requiere infraestructura real; SSE y polling se realizan fácil con json y express

**Mi elección:** Para algo personal y de pruebas yo escogería Polling , ya que es el más fácil y sencillo de hacer , pero para algo más profesional escogería el SSE ya es fácil de crear y mantener y además de que más llamativo a la vista del cliente.