

## Librerías 3D en la web / React / Next.js

Contexto general: ¿qué hay en el ecosistema 3D web?

Para renderizar contenido 3D en navegadores hoy día, lo que sucede es que todo se apoya en WebGL / WebGPU como base (o APIs similares de bajo nivel). Las librerías de más alto nivel (three.js, Babylon.js, PlayCanvas, A-Frame, etc.) ofrecen abstracciones sobre eso para que no tengas que lidiar con los shaders, buffers, contextos, etc. directamente.

Las decisiones clave al escoger una librería para tu proyecto incluyen:

- Facilidad de integración con tu stack (React, Next.js, SSR, rutas, etc.)
- Rendimiento en diferentes dispositivos (móviles, PCs, tablets)
- Ecosistema: loaders de modelos, herramientas auxiliares, comunidad
- Flexibilidad vs “todo incluido”
- Mantenimiento y documentación

Comparación técnica profunda: Three.js vs otras opciones

### Three.js como base

Ventajas de three.js:

1. Ligereza y modularidad
  - Three.js es esencialmente un “motor de renderizado” / librería de gráficos, no un engine completo con todo “preconstruido”. Esto le da modularidad: sólo usas lo que necesitas.
  - La versión “core” es más liviana que muchas comparativas de motores “todo incluido”.
  - Te permite armar tú mismo lo que necesitas: físicas, animaciones complejas, interacciones, etc.
2. Amplio ecosistema y comunidad
  - Muchas personas usan three.js, lo que implica más ejemplos, más preguntas respondidas, más herramientas auxiliares (postprocesado, loaders, addons)
  - Gran cantidad de plugins, addons “examples” oficiales, extensiones de la comunidad
  - Se puede aprovechar muchas soluciones ya probadas para shaders, efectos, animaciones, etc.
3. Máximo control
  - Al tener una “base limpia”, puedes optimizar a fondo, manejar cómo renderizas, cuándo actualizar, cómo distribuir la carga

- Puedes acceder directamente al objeto scene, cámaras, geometrías, materiales, optimizar memoria, reusar buffers, etc.
4. Compatibilidad con WebGPU / evolución futura
- La comunidad de three.js está trabajando activamente para soportar nuevas APIs gráficos.
  - No estás “atorado” en una arquitectura rígida: si necesitas “salirte” para casos muy personalizados, es más fácil.

Desventajas de three.js:

- Muchas cosas tienes que armarlas tú (física, animación avanzada, utilitarios)
- No viene con herramientas de escena, GUI, o herramientas editoras (aunque hay addons)
- Si tu proyecto escala mucho, puede requerir bastante optimización manual.

Babylon.js

Babylon.js es más cercano a un “motor 3D completo” para la web — viene con muchas cosas listas afuera del cascarón.

Ventajas de Babylon.js:

- Muchos componentes ya integrados: sistema de físicas, control de cámara, GUI dentro del canvas, armas y colisiones, etc.
- Buen soporte para realidad virtual / aumentada / WebXR
- Herramientas de debugging bastante potentes (Inspector)
- Arquitectura limpia y fuerte tipado (está basado en TypeScript)
- Puede ser más “de caja” para algunos casos donde no se quiere construirlo todo.

Desventajas de Babylon.js

- Viene con muchas cosas que puede que no necesites, lo que puede incrementar tu bundle
- Si quieres algo muy personalizado, puede que la arquitectura imponga ciertas decisiones
- En muchas comparativas, three.js “puro” queda igual o mejor si lo optimizas bien
- Algunos desarrolladores encuentran que Babylon es “más pesado de entender” cuando vas más allá de lo básico

Por ejemplo, en un artículo de LogRocket sobre “three.js vs Babylon.js”, se destaca que three es más liviano, modular y flexible, mientras que Babylon es más “todo incluido”. Otro artículo de Medium indica que Babylon es un engine más completo, mientras que Three.js es más un “renderizador + API” flexible.

### Envoltorios de React / integraciones

Para usar contenido 3D dentro de aplicaciones React (y Next.js) sin hacer cosas muy manuales, existen librerías que “envuelven” three.js:

#### @react-three/fiber

- Traduce los elementos 3D a componentes React declarativos
- Maneja el ciclo de render, redibujo, reconciliación, etc.
- Permite que uses hooks (useFrame, useThree, etc.) para animaciones y control.
- Puedes mezclar “imperativo” con “declarativo” si necesitas más control.

#### @react-three/drei

- Colección de componentes/predefinidos auxiliares para react-three-fiber
- Incluye useGLTF para cargar modelos .glb (muy usado en proyectos React)
- Contiene controles (OrbitControls, FlyControls, CameraControls), loaders, efectos, utilidades de postprocesado
- También tiene componentes como PerformanceMonitor que pueden ajustar la calidad dependiendo del desempeño.
- En su repositorio oficial se presenta como “una colección de helpers y abstracciones funcionales listas para usar con React Three Fiber”

Estado ligero (por ejemplo, zustand u otros)

- Si la escena 3D necesita intercambiar datos con UI (por ejemplo, un menú que cambia el color del objeto 3D, activar animaciones, etc.), conviene tener un store global.
- Zustand es muy minimalista, sin mucha sobrecarga, y se puede usar para conectar estado React con lógica 3D sin re-renderes innecesarios.
- En la documentación de react-three-fiber se advierte que no se haga actualizaciones de estado rápido dentro de loops de render (ej. dentro de useFrame) porque eso puede afectar el rendimiento. En esos casos es mejor mutar directamente o leer del store.

### Justificación final y recomendaciones

Con toda esta información, la decisión de usar three + react-three-fiber + drei se refuerza porque:

- Te da la mejor combinación entre control y productividad: tienes el motor potente (three) y una forma cómoda de integrarlo en React (fiber), además de utilidades (drei).
- Te permite comenzar pequeño pero escalar (puedes mejorar la escena, añadir físicas, optimizar, etc.).
- La comunidad y documentación son fuertes, lo cual ayuda cuando te atoras.
- Te pone “el poder en tus manos” — si más adelante decides cambiar partes del pipeline, no estás pegado a una arquitectura rígida.

Para visualización, productos 3D, renders interactivos, galerías, y lo que normalmente se hace en aplicaciones web “normales con 3D extra”, esta pila es una de las opciones preferidas hoy.