🛠️ React DevTools

🎯 Contenido

🌟 1. Introducción a React DevTools

React Developer Tools es una herramienta oficial mantenida por el equipo de React. Su propósito es inspeccionar, depurar y optimizar aplicaciones React directamente desde el navegador.

React Developer Tools es **obligatorio** para cualquier developer de React. Si no lo estás usando, básicamente estás programando a ciegas con respecto a la estructura de componentes, el flujo de estado y el rendimiento.

📥 2. Instalación

Instala el complemento de Chrome y cuando uses una página con React aparecerán las pestañas **Components** y **Profilers** (Cierra y vuelve a abrir el navegador en el caso de que no aparezca).

Páginas para probar: https://react.dev y https://www.airbnb.com/

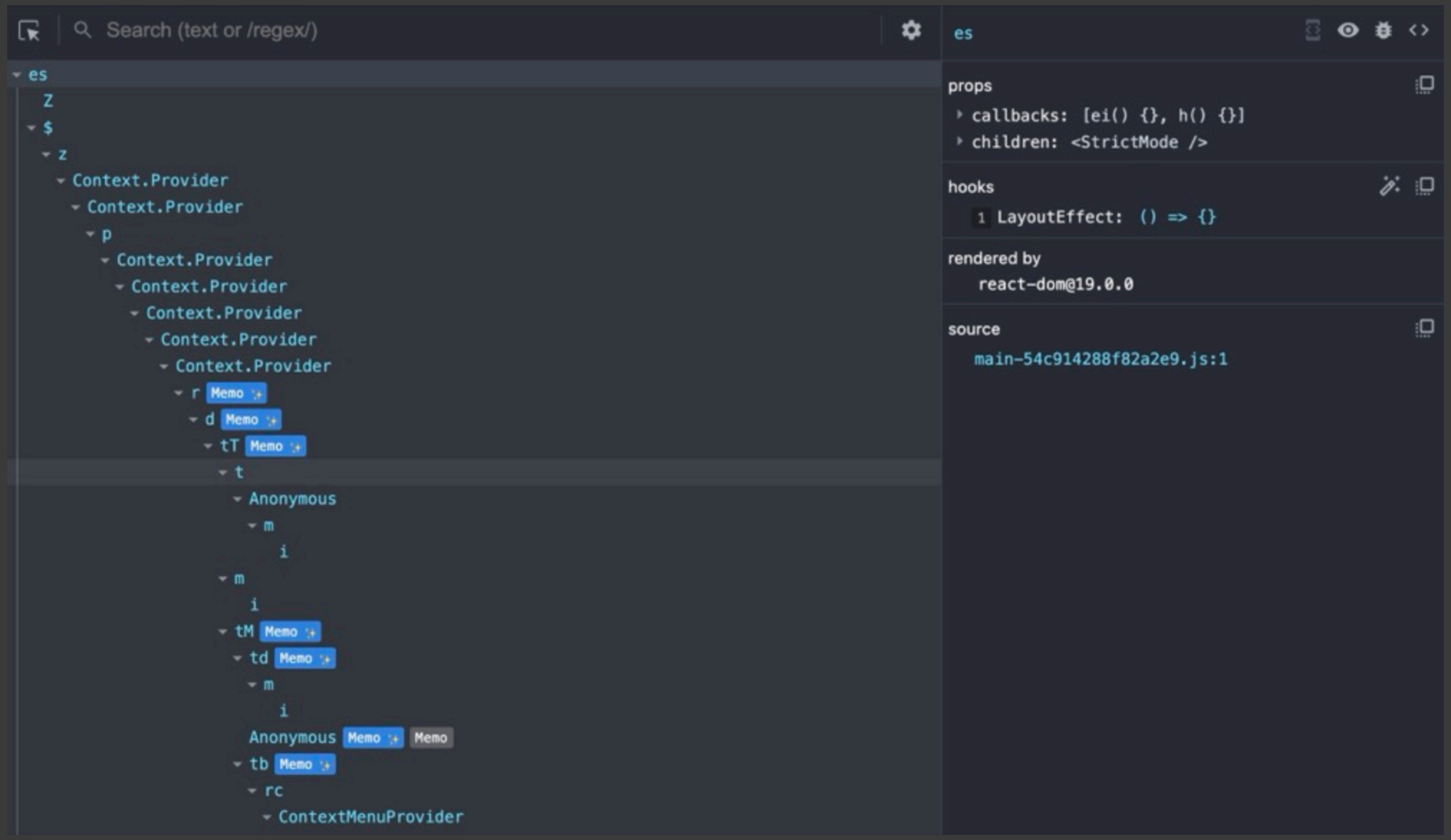
> ¿Has instalado el complemento? ¿Has verificado que funciona en alguna página con React? (El profiler funciona en modo dev)

🔍 3. Components Panel

Funcionalidades principales:

- 🌳 Ver la jerarquía completa de componentes React (no solo nodos DOM)
- ✏️ Editar state y props en vivo (debugging, no se persisten)
- 🎯 Identificar instantáneamente componentes container vs presentacionales:
  - **Hooks section** → 🧠 Container
  - **Solo Props, sin hooks** → 🎨 Presentacional

> ¿Has inspeccionado la estructura de componentes de alguna app React? ¿Has editado props o state en vivo? ¿Puedes distinguir componentes container de presentacionales?


Search (text or /regex/)
es
Z
$
z
Context.Provider
Context.Provider
p
Context.Provider
Context.Provider
Context.Provider
Context.Provider
Context.Provider
r Memo
d Memo
tT Memo
t
Anonymous
m
i
m
i
tM Memo
td Memo
m
i
Anonymous Memo Memo
tb Memo
rc
ContextMenuProvider
es
props
callbacks: [ei() {}, h() {}]
children: <StrictMode />
hooks
1 LayoutEffect: () => {}
rendered by
react-dom@19.0.0
source
main-54c914288f82a2e9.js:1

📊 4. Profiler Panel

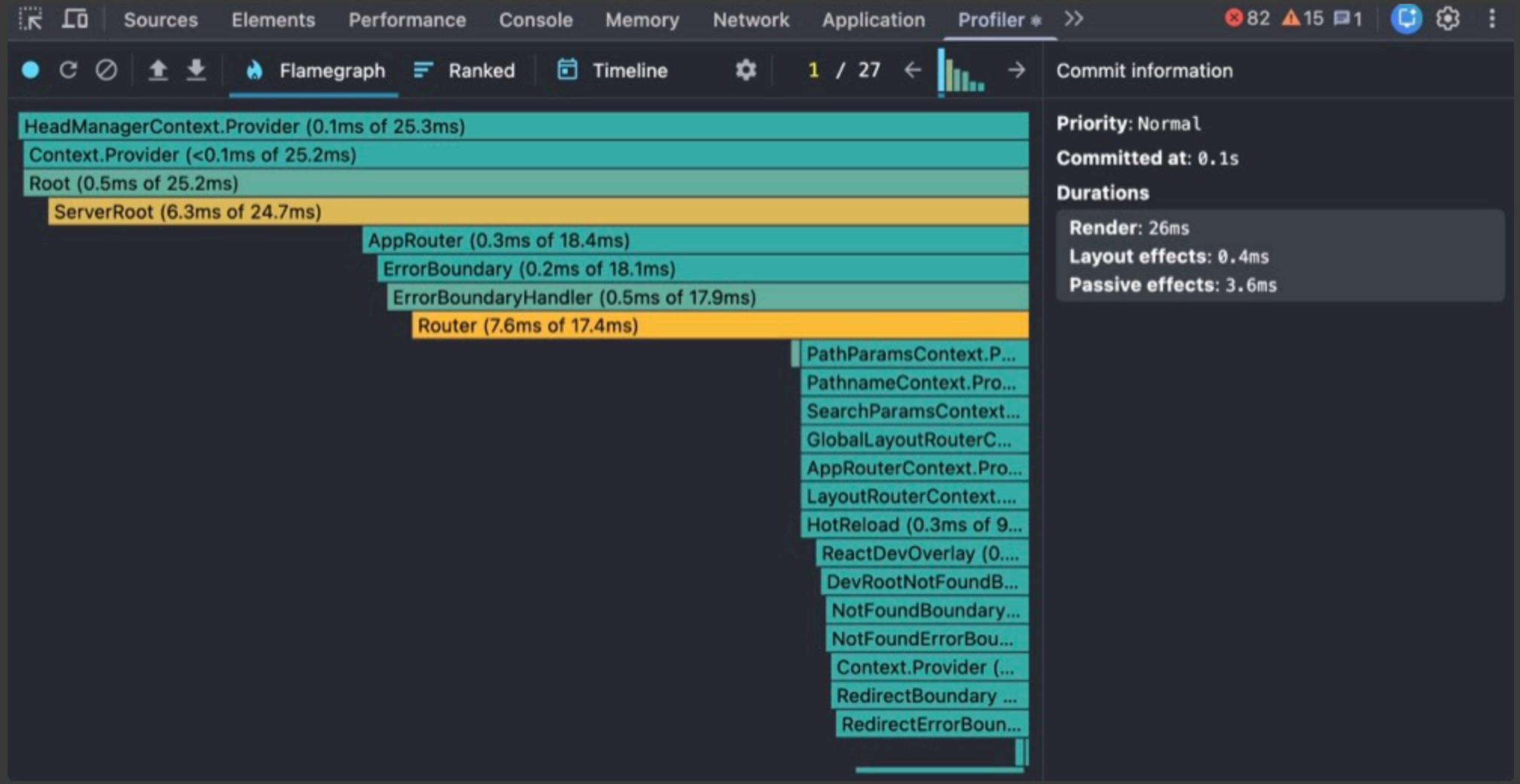
> ⚠️ **Importante**: Tiene que estar en modo dev para que funcione. Probar en local en modo dev.

Puedes grabar el comportamiento y ver sus renderizados. Grabas y recargas desde el profiler.

Análisis disponible:

- 📈 **Gráfico de carga**: Visualización temporal de renders
- 🏆 **Ranking de tiempo**: Durations

> ¿Has usado el Profiler en modo dev? ¿Has grabado una sesión y analizado los tiempos de renderizado?


Sources
Elements
Performance
Console
Memory
Network
Application
Profiler
82
15
1
Flamegraph
Ranked
Timeline
1 / 27
Commit information
HeadManagerContext.Provider (0.1ms of 25.3ms)
Context.Provider (<0.1ms of 25.2ms)
Root (0.5ms of 25.2ms)
ServerRoot (6.3ms of 24.7ms)
AppRouter (0.3ms of 18.4ms)
ErrorBoundary (0.2ms of 18.1ms)
ErrorBoundaryHandler (0.5ms of 17.9ms)
Router (7.6ms of 17.4ms)
PathParamsContext.P...
PathnameContext.Pro...
SearchParamsContext...
GlobalLayoutRouterC...
AppRouterContext.Pro...
LayoutRouterContext....
HotReload (0.3ms of 9...
ReactDevOverlay (0....
DevRootNotFoundB...
NotFoundBoundary...
NotFoundErrorBou...
Context.Provider (...
RedirectBoundary ...
RedirectErrorBoun...
Priority: Normal
Committed at: 0.1s
Durations
Render: 26ms
Layout effects: 0.4ms
Passive effects: 3.6ms

Métricas clave del Profiler:

- ⏱️ **Render time**: Tiempo en renderizarse. Te ayuda a identificar componentes lentos que necesitan optimización (memo, useMemo, etc.). Render no cambia la pantalla: solo es cálculo.
- 🔄 **Commit**: Cada "commit" representa una actualización completa de la UI. React agrupa todos los cambios (Renders) en un solo commit para actualizar el DOM de forma eficiente. Puedes ver cuántos commits ocurren durante una interacción. Render es barato commit es caro
- 🎭 **Passive effects**: Efectos que se ejecutan después del renderizado (como `useEffect`). Se ejecutan en el orden: Render → Commit → Paint → Passive Effects. Útil para identificar efectos costosos que pueden estar bloqueando la UI.

⚙️ 5. Configuración y Tips

Highlight component updates: En ⚙️ Settings. Check: ☑️ Highlight updates when components render

Esta opción resalta visualmente qué componentes se están renderizando en tiempo real, lo cual es muy útil para identificar renders innecesarios.

> ¿Has activado el highlight de actualizaciones? ¿Has identificado componentes que se renderizan más de lo necesario?

🎉 ¡Gracias!

Lo que no se evalúa se devalúa

Lo que se mide, se puede mejorar

Acostúmbrate a auditarte 😉