











# Contenido

1.  Introducción a Chrome DevTools
2.  Abrir DevTools
3.  Elements Panel
4.  Console Panel
5.  Network Panel
6.  Lighthouse
7.  Application Panel
8.  Tips Avanzados

# 🌟 1. Introducción a Chrome DevTools




Chrome DevTools es un conjunto de herramientas de desarrollo web integradas en Google Chrome que nos permite ver por dentro todas las webs. Entre muchas cosas podemos hacer:

- 🔍 Inspeccionar y modificar el código en tiempo real (No se guarda)
- 🐛 Depurar problemas rápidamente
- 📊 Analizar el rendimiento
- 📱 Simular dispositivos móviles

| El resto de navegadores tienen una aplicación parecida

## 2. Abrir DevTools




Hay varias formas de acceder a las DevTools:

-  Acceso rápido:
  - Mac: `Cmd + Opt + I`
  - Windows/Linux: `F12` o `Ctrl + Shift + I`
-  Encima de un elemento: Click derecho → "Inspeccionar"
-  Dock positions: Abajo, derecha, separada, etc.

| ¿Tenemos abierto el inspector de elementos?

## 3. Elements Panel




Herramienta principal para inspeccionar y modificar el DOM:

-  Inspeccionar y modificar HTML en tiempo real. (Cambiar el título)
-  Ver y editar CSS:
  - Añadir/borrar propiedades (colores, alineaciones, tamaños, etc.)
  - Panel Computed: Ver todas las capas de CSS aplicadas e ir a la que aplica
  - Experimentar con Flexbox/Grid (En apple.com el ul del nav puedo ajustar)
  - Control de animaciones ([Ver en W3S](#)) Activar en los circulitos
-  Estados de elementos: Pseudoselectores y pseudoclasas (:hover, :active) (En apple.com forzar el estado del botton buy al hover)

| ¿Has editado algo de alguna página?

## 4. Console Panel






Centro de diagnóstico y debugging:

-  Ejecutar JavaScript en vivo: `alert('Hola');` `body = document.getElementsByTagName('body')`
-  Ver errores y warnings
-  Usar `debugger;` para pausar la ejecución, con temporizado:  
`setTimeout(function(){ debugger; }, 3000);`

| ¿Has ejecuta algo de JS? ¿Has visto la consola de instagram?

## 5. Network Panel

Analiza el rendimiento de red:

-  Timeline de requests
-  Tamaño de recursos
-  Tiempos de carga
-  Simular conexiones lentas con Throttling, simular sin caché
-  Detalles de cada petición. (Por ejemplo para ver qué está devolviendo el back y a veces quiero ver el Preview)

¿Has verificado el tiempo real de carga de una página sin cache y estrangulando la conexión?

## 6. Lighthouse

Diferentes tipos de auditoría, te dice en qué falla y referencias para solucionar el problema:

- Performance
- Accessibility
- Best practices
- SEO

| ¿Has auditado una página web?










## 7. Application Panel

Gestión de datos del lado del cliente (Antes de que el usuario cree una cuenta podemos guarda ry recuperar información del usuario):

-  Cookies (Revisar instalación y gestión. Antes de meter nada debería de avisar)
-  Local Storage vs Session Storage (Diferencia clave: Local persiste a la sesión)
-  Session Storage

| ¿Revisa que no te hayan instalado nada que no hayas autorizado?

## 8. Tips Avanzados

-  Practica con el modo responsive
-  TODO: Modificar sensores para testing
-  TODO: Verificar accesibilidad: Medir contraste de colores y Auditorías y recomendaciones
-  TODO: Coverage: Encuentra CSS y JS no utilizado: Analiza código cargado vs utilizado y Optimiza el tamaño de tus páginas
-  TODO: Progressive Web Apps
-  TODO: Personaliza tu workspace
-  TODO: Aprende las Chrome DevTools APIs

 ¡Gracias!