

Frontend roadmap.sh

Internet

Internet es una red mundial de ordenadores conectados entre sí que se comunican mediante un conjunto normalizado de protocolos.

Visite los siguientes recursos para obtener más información:

- [¿Cómo funciona Internet?](#)
- [Explicación de Internet](#)
- [¿Cómo funciona Internet?](#)
- [Introducción a Internet](#)
- [Aprenda cómo funciona la Web](#)
- [¿Cómo funciona Internet?](#)
- [Cómo funciona Internet en 5 minutos](#)

¿Cómo funciona Internet?

Internet es una red mundial de ordenadores conectados entre sí que se comunican mediante un conjunto normalizado de protocolos.

Visite los siguientes recursos para saber más:

- [¿Cómo funciona Internet?](#)
- [¿Cómo funciona Internet? Documentos MDN](#)
- [Introducción a Internet](#)
- [¿Cómo funciona Internet?](#)
- [Cómo funciona Internet en 5 minutos](#)

¿Qué es HTTP?

HTTP es el protocolo de comunicación de capa de aplicación basado en **TCP/IP** que estandariza la forma en que el cliente y el servidor se comunican entre sí. HTTP sigue un "modelo cliente-servidor" clásico, en el que el cliente abre una solicitud de conexión y espera a recibir una respuesta. HTTP es un protocolo sin estado, lo que significa que el servidor no conserva ningún dato (estado) entre dos peticiones.

Visite los siguientes recursos para obtener más información:

- [Todo lo que necesitas saber sobre HTTP](#)
- [¿Qué es HTTP?](#)
- [Cómo funciona HTTPS ...¡en un cómic!](#)
- [Visión general de HTTP](#)
- [HTTP/3 de la A a la Z: Conceptos básicos](#)
- [HTTP/3 ya es un estándar: Por qué usarlo y cómo empezar](#)
- [Curso acelerado y exploración de HTTP](#)

¿Qué es un nombre de dominio?

Un nombre de dominio es una dirección única y fácil de recordar que se utiliza para acceder a sitios web, como "google.com" y "facebook.com". Los usuarios pueden conectarse a sitios web utilizando nombres de dominio gracias al Sistema de Nombres de Dominio (DNS).

Visite los siguientes recursos para obtener más información:

- [¿Qué es un nombre de dominio?](#)
- [¿Qué es un nombre de dominio? | Nombre de dominio frente a URL](#)
- [Guía para principiantes sobre el funcionamiento de los nombres de dominio](#)

¿Qué es el alojamiento web?

El alojamiento web es un servicio en línea que le permite publicar los archivos de su sitio web en Internet. Así, cualquiera que tenga acceso a Internet podrá acceder a su sitio web.

Visite los siguientes recursos para obtener más información:

- [¿Qué es el alojamiento web? Explicación](#)
- [Explicación de los distintos tipos de alojamiento web](#)
- [Dónde alojar un proyecto Fullstack con poco presupuesto](#)

¿Cómo funciona el DNS?

El Sistema de Nombres de Dominio (DNS) es la guía telefónica de Internet. Los humanos acceden a la información en línea a través de nombres de dominio, como nytimes.com o espn.com. Los navegadores web interactúan a través de direcciones IP (Protocolo de Internet). El DNS traduce los nombres de dominio a direcciones IP para que los navegadores puedan cargar los recursos de Internet.

Visite los siguientes recursos para obtener más información:

- [¿Qué es el DNS?](#)
- [Enredar con DNS - DNS Playground](#)
- [Cómo funciona el DNS \(cómic\)](#)
- [DNS y su funcionamiento](#)
- [Registros DNS](#)
- [Cuando añadir registros de cola a la configuración DNS](#)
- [Registros DNS para novatos - Cómo gestionar los registros de un sitio web](#)

Navegadores y su funcionamiento

Un navegador web es una aplicación de software que permite a un usuario acceder y visualizar páginas web u otros contenidos en línea a través de su interfaz gráfica de usuario.

Visite los siguientes recursos para obtener más información:

- [Cómo funcionan los navegadores](#)
- [Función del motor de renderizado en los navegadores](#)

- [Rellenando la Página: Cómo funcionan los navegadores](#)
- [¿Cómo funcionan los navegadores?](#)

HTML

HTML son las siglas de HyperText Markup Language. Se utiliza en el frontend y da estructura a la página web, a la que se puede dar estilo con CSS y hacer interactiva con JavaScript.

Visite los siguientes recursos para obtener más información:

- [W3Schools: Aprender HTML](#)
- [htmlreference.io: Todos los elementos HTML de un vistazo](#)
- [HTML para principiantes](#)
- [Conceptos básicos de desarrollo web](#)
- [Codecademy - Aprende HTML](#)
- [Curso Interactivo de HTML](#)
- [Curso Completo de HTML para Principiantes | Tutorial Completo Todo en Uno](#)
- [Curso Completo de HTML - Tutorial para Construir un Sitio Web](#)
- [Tutorial de HTML para Principiantes: Curso acelerado de HTML](#)

Aprender lo básico

HTML son las siglas de HyperText Markup Language. Se utiliza en el frontend y da estructura a la página web, a la que se puede dar estilo con CSS y hacer interactiva con JavaScript.

Visite los siguientes recursos para obtener más información:

- [W3Schools: Aprender HTML](#)
- [MDN Docs: Primeros pasos con HTML](#)
- [web.dev: Aprende HTML](#)
- [Curso completo de HTML - Tutorial para crear un sitio web](#)
- [Tutorial de HTML para principiantes: Curso acelerado de HTML](#)
- [HTML Cheatsheet](#)
- [Scaler: HTML](#)

Escribir HTML semántico

Un elemento semántico describe claramente su significado tanto para el navegador como para el desarrollador. En HTML, los elementos semánticos son el tipo de elementos que se pueden utilizar para definir diferentes partes de una página web, como `<form>`, `<table>`, `<article>`, `<header>`, `<footer>`, etc.

Visite los siguientes recursos para obtener más información:

- [Guía para escribir HTML semántico](#)
- [W3Schools: HTML Semántico](#)
- [Cómo escribir HTML semántico](#)
- [HTML Best Practices - Cómo construir un mejor sitio web basado en HTML](#)
- [HTML semántico: Qué es y cómo mejora su sitio web](#)
- [Marcado semántico](#)
- [HTML semántico - web.dev](#)

Formularios y validaciones

Antes de enviar los datos al servidor, es importante asegurarse de que todos los controles de formulario requeridos se rellenan en el formato correcto. Esto se denomina validación de formularios del lado del cliente, y ayuda a garantizar que los datos enviados coinciden con los requisitos establecidos en los distintos controles del formulario.

Visite los siguientes recursos para obtener más información:

- [MDN Web Docs: Validación de formularios del lado del cliente](#)
- [Aprender formularios por web.dev](#)
- [W3Schools: Validación de formularios en JavaScript](#)

Accesibilidad

La accesibilidad web significa que los sitios web, las herramientas y las tecnologías se diseñan y desarrollan de forma que las personas con discapacidad puedan utilizarlos fácilmente.

Visite los siguientes recursos para saber más:

- [Desarrollo para la accesibilidad web mediante WAI del W3C](#)
- [Tutorial de accesibilidad](#)
- [Guía completa de componentes de front-end accesibles](#)
- [Lista de reproducción completa sobre accesibilidad](#)
- [Accesibilidad MDN](#)
- [Accesibilidad para desarrolladores de Google](#)
- [Accesibilidad web de Udacity](#)
- [La accesibilidad como parte esencial de la experiencia inclusiva del desarrollador](#)

Conceptos básicos de SEO

SEO o Search Engine Optimization es la técnica utilizada para optimizar su sitio web para una mejor clasificación en los motores de búsqueda como Google, Bing, etc.

Visite los siguientes recursos para obtener más información:

- [Google Search Central - Documentos SEO](#)
- [Guía SEO](#)
- [8 prácticas recomendadas de SEO que los desarrolladores deben conocer](#)
- [SEO para desarrolladores](#)
- [Curso completo de SEO para principiantes](#)
- [Curso de SEO para expertos](#)
- [Aprender SEO](#)

CSS

CSS u hojas de estilo en cascada (Cascading Style Sheets) es el lenguaje utilizado para dar estilo al frontend de cualquier sitio web. CSS es una tecnología fundamental de la World Wide Web, junto con HTML y JavaScript.

Visite los siguientes recursos para obtener más información:

- [The Odin Project](#)
- [¡Qué es Flexbox!](#)
- [Aprende CSS | Codecademy](#)
- [Aprende CSS Intermedio | Codecademy](#)
- [CSS Curso Completo](#)
- [CSS Curso Intensivo Para Principiantes Absolutos](#)
- [Tutorial de HTML y CSS](#)
- [CSS Masterclass - Tutorial y Curso para Principiantes](#)
- [W3Schools - Aprende CSS](#)
- [cssreference.io: Todas las propiedades CSS de un vistazo](#)
- [Web.dev by Google - Aprender CSS](#)
- [freeCodeCamp - Diseño web responsivo](#)
- [Aprende a programar HTML y CSS](#)
- [Entradas del blog CSS Hack de Joshw Comeaus](#)
- [100 Días CSS Challenge](#)
- [CSS Tutorial | Scaler](#)

Aprender lo básico

CSS u hojas de estilo en cascada (Cascading Style Sheets) es el lenguaje utilizado para dar estilo al frontend de cualquier sitio web. CSS es una tecnología fundamental de la World Wide Web, junto con HTML y JavaScript.

Visite los siguientes recursos para obtener más información:

- [W3Schools - Aprende CSS](#)
- [web.dev - Aprende CSS](#)
- [freeCodeCamp - Diseño Web Responsive](#)
- [Aprende a codificar HTML y CSS](#)
- [Curso acelerado de CSS para principiantes absolutos](#)
- [Tutorial de HTML y CSS](#)
- [CSS Masterclass - Tutorial y Curso para Principiantes](#)

Hacer diseños

Float, grid, flexbox, positioning, display y box model son algunos de los temas clave que se utilizan para hacer layouts. Utiliza los siguientes recursos para aprender sobre estos temas:

Visita los siguientes recursos para aprender más:

- [Aprende y practica Flexbox](#)

- [Juego para aprender CSS Grid](#)
- [Todo sobre Floats](#)
- [Tipos de Posicionamiento: ¿En qué se diferencian?](#)
- [El Box Model](#)
- [La Propiedad CSS Display](#)
- [Guía completa de Flexbox](#)
- [Guía completa de Grid](#)
- [Aprende CSS Grid - Curso](#)
- [Aprende CSS Grid gratis](#)
- [Aprende por último Grid con el módulo CSS Grid Layout](#)

Diseño web adaptable

El diseño web adaptable es la técnica que permite que las páginas web se vean bien en todos los tamaños de pantalla. Para ello se utilizan determinadas técnicas, como las consultas de medios CSS, los porcentajes de anchura, las alturas mínimas o máximas de anchura, etc.

Visite los siguientes recursos para obtener más información:

- [Diseño Web Responsive](#)
- [Aprenda diseño responsivo](#)
- [Guía para principiantes del diseño web con capacidad de respuesta](#)
- [La guía del diseño web responsive en 2022](#)
- [5 sencillos consejos para hacer diseños responsive de forma fácil](#)
- [Introducción al diseño web responsivo](#)

JavaScript

JavaScript le permite añadir interactividad a sus páginas. Ejemplos comunes que puede haber visto en los sitios web son sliders, interacciones de clic, popups, etc.

Visite los siguientes recursos para obtener más información:

- [Visite la hoja de ruta dedicada a JavaScript](#)
- [W3Schools - Tutorial de JavaScript](#)
- [El Tutorial de JavaScript Moderno](#)
- [Aprenda JavaScript: Cubre muchos temas](#)
- [Libro de texto Eloquent JavaScript](#)
- [You Dont Know JS Yet \(serie de libros\)](#)
- [Curso intensivo de JavaScript para principiantes](#)
- [Construye un clon de la página de aterrizaje de Netflix con HTML, CSS y JS](#)
- [Construye 30 proyectos Javascript en 30 días](#)
- [Aprenda los conceptos básicos de JavaScript](#)
- [JavaScript para principiantes](#)

Aprender lo básico

JavaScript le permite añadir interactividad a sus páginas. Ejemplos comunes que puede haber visto en los sitios web son sliders, interacciones de clic, popups, etc.

Visite los siguientes recursos para obtener más información:

- [W3Schools - Tutorial de JavaScript](#)
- [El Tutorial Moderno de JavaScript](#)
- [Curso intensivo de JavaScript para principiantes](#)
- [Crea un clon de la página de aterrizaje de Netflix con HTML, CSS y JS](#)

Manipulación del DOM

El Modelo de Objetos del Documento (DOM) es una interfaz de programación construida para documentos HTML y XML. Representa la página que permite a los programas y scripts actualizar dinámicamente la estructura, el contenido y el estilo del documento. Con DOM, podemos acceder y manipular fácilmente etiquetas, IDs, clases, atributos, etc.

Visite los siguientes recursos para obtener más información:

- [Árbol DOM](#)
- [GeeksForGeeks - DOM \(Modelo de Objetos del Documento\)](#)
- [¿Qué es el DOM?](#)
- [JavaScript Eloquent, 3ª Edición: El Modelo de Objetos del Documento](#)
- [JavaScript HTML DOM](#)
- [JavaScript DOM](#)
- [Aprenda el HTML DOM con ejercicios - CodeGuage](#)
- [¿Qué es DOM, Shadow DOM y Virtual DOM?](#)
- [Curso intensivo de JavaScript DOM](#)

Fetch API / Ajax (XHR)

Ajax es la técnica que nos permite enviar y recibir datos de forma asíncrona desde los servidores, por ejemplo, actualizar el perfil de usuario o recuperar de forma asíncrona la lista de productos buscados sin recargar la página.

Visite los siguientes recursos para obtener más información:

- [API Fetch MDN Docs](#)
- [Guía sencilla de la API Fetch de JavaScript](#)
- [Introducción a Fetch](#)
- [API de obtención de JavaScript](#)

Sistemas de control de versiones

Los sistemas de control de versiones le permiten realizar un seguimiento de los cambios en su código base/archivos a lo largo del tiempo. Te permiten volver a una versión anterior del código sin ningún problema. Además, ayudan a colaborar con otras personas que trabajan en el mismo código: si alguna vez has colaborado con otras personas en un proyecto, es posible que ya conozcas la frustración de copiar y fusionar los cambios de otra persona en tu código base; los sistemas de control de versiones te permiten deshacerte de este problema.

Visite los siguientes recursos para obtener más información:

- [Introducción al sistema de control de versiones](#)
- [Curso acelerado de Git y GitHub para principiantes](#)
- [Aprende Git en 20 minutos](#)
- [Documentación de Git](#)
- [Aprende Git con Atlassian](#)

Git

Git es un sistema de control de versiones distribuido, gratuito y de código abierto, diseñado para gestionar con rapidez y eficacia desde proyectos pequeños a muy grandes.

Visita los siguientes recursos para obtener más información:

- [Curso acelerado de Git y GitHub para principiantes](#)
- [Aprende Git con tutoriales, noticias y consejos - Atlassian](#)
- [Git Cheatsheet](#)
- [Tutorial: Git para absolutamente todo el mundo](#)

Servicios de alojamiento de repositorios

Existen diferentes servicios de alojamiento de repositorios, siendo los más famosos GitHub, GitLab y BitBucket. Yo recomendaría crear una cuenta en GitHub porque es donde se realiza la mayor parte del trabajo OpenSource y donde están la mayoría de los desarrolladores.

Visita los siguientes recursos para obtener más información:

- [GitHub: Donde el mundo crea software](#)
- [GitLab: Iterar más rápido, innovar juntos](#)
- [BitBucket: La solución Git para equipos profesionales](#)

GitHub

GitHub es un proveedor de alojamiento en Internet para el desarrollo de software y el control de versiones mediante Git. Ofrece las funcionalidades de control de versiones distribuidas y gestión de código fuente de Git, además de sus propias características.

Visita los siguientes recursos para obtener más información:

- [GitHub: Inicio rápido](#)
- [Aprende GitHub](#)
- [¿Qué es GitHub?](#)
- [Git frente a GitHub: ¿Cuál es la diferencia?](#)
- [Git y GitHub para principiantes](#)
- [Git y GitHub - CS50 más allá de 2019](#)

GitLab

GitLab es un proveedor de alojamiento en Internet para el desarrollo de software y el control de versiones mediante Git. Ofrece las funcionalidades de control de versiones distribuidas y gestión de código fuente de Git, además de sus propias características.

Visita los siguientes recursos para obtener más información:

- [Documentación de GitLab](#)
- [Sitio web de GitLab](#)
- [Desarrollo: Conectar git a GitLab para pequeños proyectos](#)

BitBucket

BitBucket es un proveedor de alojamiento en Internet para desarrollo de software y control de versiones mediante Git. Ofrece las funcionalidades de control de versiones distribuidas y gestión de código fuente de Git, además de sus propias características.

Visita los siguientes recursos para obtener más información:

- [¿Cómo utilizar BitBucket?](#)

- [Sitio web de BitBucket](#)

Gestores de paquetes

Los gestores de paquetes te permiten gestionar las dependencias (código externo escrito por ti o por otra persona) que tu proyecto necesita para funcionar correctamente.

Visita los siguientes recursos para obtener más información:

- [JavaScript moderno para dinosaurios](#)
- [Guía para principiantes sobre el uso de npm](#)
- [Yarn - Primeros pasos](#)
- [Tutorial de NPM para principiantes](#)
- [Curso intensivo de NPM](#)
- [Curso intensivo de Yarn](#)

npm

npm es un gestor de paquetes para el lenguaje de programación JavaScript mantenido por npm, Inc. npm es el gestor de paquetes por defecto para el entorno de ejecución JavaScript Node.js.

Visite los siguientes recursos para obtener más información:

- [JavaScript moderno para dinosaurios](#)
- [Guía para principiantes sobre el uso de npm](#)
- [Cómo usar NPM](#)
- [Tutorial de NPM para principiantes](#)
- [Curso acelerado de NPM](#)

pnpm

PNPM es un gestor de paquetes alternativo para Node.js que significa "Performant NPM". El objetivo principal de PNPM es mantener todos los paquetes en un almacén global (centralizado) y utilizarlos si otros proyectos también los necesitan creando enlaces directos a él.

Visita los siguientes recursos para saber más:

- [Sitio web oficial](#)
- [Conozca PNPM: El PNPM más rápido y eficaz](#)

Yarn

Yarn es un sistema de empaquetado de software desarrollado en 2016 por Facebook para el entorno de ejecución JavaScript Node.js que proporciona velocidad, consistencia, estabilidad y seguridad como alternativa a npm (gestor de paquetes).

Visita los siguientes recursos para obtener más información:

- [JavaScript moderno para dinosaurios](#)
- [Yarn - Primeros pasos](#)

- [Curso intensivo de Yarn](#)

Elige un framework

Los frameworks web están diseñados para escribir aplicaciones web. Los frameworks son colecciones de bibliotecas que ayudan en el desarrollo de un producto de software o sitio web. Los frameworks para el desarrollo de aplicaciones web son colecciones de varias herramientas. Los marcos varían en sus capacidades y funciones, dependiendo de las tareas establecidas. Definen la estructura, establecen las reglas y proporcionan las herramientas de desarrollo necesarias.

Visite los siguientes recursos para obtener más información:

- [¿Cuál es la diferencia entre un framework y una librería?](#)
- [¿Qué framework JS es mejor?](#)

React

React es la biblioteca JavaScript más popular para la creación de interfaces de usuario. React también puede renderizar en el servidor utilizando Node y potenciar aplicaciones móviles utilizando React Native.

Visite los siguientes recursos para obtener más información:

- [Visite la hoja de ruta dedicada de React](#)
- [Sitio web de React](#)
- [Introducción oficial](#)
- [Guía de React para principiantes](#)
- [Curso de React JS para principiantes](#)
- [Curso de React - Tutorial para principiantes sobre la biblioteca JavaScript de React \[2022\]](#)
- [Entendiendo el Proceso de Renderizado de la UI de React](#)

Vue.js

Vue.js es un framework JavaScript de código abierto para crear interfaces de usuario y aplicaciones de una sola página. Se centra principalmente en el desarrollo front-end.

Visite los siguientes recursos para obtener más información:

- [Visite la hoja de ruta dedicada de Vue](#)
- [Sitio web de Vue.js](#)
- [Introducción oficial](#)
- [Curso de Vue.js para principiantes](#)
- [Curso intensivo de Vue.js](#)
- [Conozca Vue.js, el framework JavaScript flexible](#)

Angular

Angular es un marco de desarrollo front-end basado en componentes construido sobre TypeScript que incluye una colección de librerías bien integradas que incluyen características como enrutamiento, gestión de formularios, comunicación cliente-servidor y mucho más.

Visite los siguientes recursos para obtener más información:

- [Visita Dedicated Angular Roadmap](#)
- [Oficial - Primeros pasos con Angular](#)
- [Curso de Angular para principiantes \[Tutorial completo de Front End con TypeScript\]](#)

Svelte

Svelte es un framework javascript que a diferencia de Vue y React no utiliza DOM diffing vertical sino que sabe exactamente qué y dónde actualizar cuando el estado cambia. Se centra principalmente en frontend y la construcción de interfaces de usuario.

Visita los siguientes recursos para aprender más:

- [Sitio web de Svelte](#)
- [Documentación de Svelte](#)
- [Lista de reproducción del curso de Svelte para principiantes](#)
- [Todo sobre Svelte, el marco de trabajo web basado en estados más apreciado](#)
- [Svelte y el futuro del desarrollo Frontend](#)

SolidJS

Solid es un conjunto de herramientas de JavaScript reactivo para crear interfaces de usuario sin un DOM virtual. Para garantizar que sólo se ejecuta el código relevante cuando se produce una actualización de estado, compila las plantillas en nodos DOM reales una vez y envuelve las modificaciones en reacciones detalladas.

Visite los siguientes recursos para obtener más información:

- [Official Website - SolidJS](#)
- [Tutorial de SolidJS](#)

Qwik

Qwik es un nuevo tipo de framework web que puede ofrecer aplicaciones web de carga instantánea de cualquier tamaño o complejidad. Sus sitios y aplicaciones pueden arrancar con aproximadamente 1kb de JS (independientemente de la complejidad de la aplicación), y lograr un rendimiento constante a escala.

Más información sobre Qwik **en el sitio web oficial**.

CSS moderno

La forma en que escribimos CSS en nuestras aplicaciones front-end modernas es completamente diferente de cómo solíamos escribir CSS antes. Existen métodos como Styled Components, CSS Modules, Styled JSX, Emotion, etc.

Tailwind CSS

Marco CSS que proporciona clases CSS atómicas para ayudarte a dar estilo a componentes como `flex`, `pt-4`, `text-center` y `rotate-90` que pueden componerse para construir cualquier diseño, directamente en tu marcado.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Tailwind](#)
- [Tailwind CSS Curso Completo para Principiantes](#)
- [Curso intensivo de CSS de Tailwind](#)
- [¿Deberías usar Tailwind CSS?](#)
- [Screencasts oficiales](#)

Radix UI

Una biblioteca de componentes de interfaz de usuario de código abierto para crear sistemas de diseño y aplicaciones web accesibles y de alta calidad.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentación oficial](#)

Shadcn UI

Shadcn UI son componentes bellamente diseñados que puede copiar y pegar en sus aplicaciones. Accesibles. Personalizables. De código abierto. Están contruidos con Tailwind CSS y Radix UI.

- [Sitio web oficial](#)
- [Repositorio oficial GitHub](#)

Arquitectura CSS

CSS es notoriamente difícil de manejar en sistemas grandes, complejos y de rápida iteración. Existen diferentes formas de escribir CSS que permiten escribir CSS más mantenibles.

Visite los siguientes recursos para obtener más información:

- [Algunas metodologías CSS](#)
- [Sitio web oficial de BEM](#)
- [Sitio web oficial de OOCSS](#)
- [Sitio web oficial de SMACSS](#)

BEM

La metodología Block, Element, Modifier (comúnmente conocida como BEM) es una popular convención de nomenclatura para clases en HTML y CSS. Desarrollada por el equipo de Yandex, su objetivo es ayudar a los desarrolladores a comprender mejor la relación entre el HTML y el CSS en un proyecto determinado.

Visita los siguientes recursos para obtener más información:

- [Página oficial de BEM](#)
- [Documentación BEM](#)
- [BEM 101](#)
- [Tutoriales de BEM](#)

Preprocesadores CSS

Los preprocesadores de CSS son lenguajes de programación que amplían las capacidades predeterminadas de CSS. Nos permiten utilizar la lógica en nuestro código CSS, como variables, anidamiento, herencia, mixins, funciones y operaciones matemáticas.

Visite los siguientes recursos para obtener más información:

- [Explicación de los preprocesadores CSS](#)
- [¿Por qué utilizar preprocesadores?](#)

Sass

Sass es un lenguaje de scripting preprocesador que se interpreta o compila en Hojas de Estilo en Cascada. Permite escribir CSS mantenibles y proporciona características como variables, anidamiento, mixins, extensión, funciones, bucles, condicionales, etc.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Sass](#)
- [Documentación oficial](#)
- [Tutorial de Sass para principiantes](#)
- [Sass, BEM y diseño responsivo](#)

PostCSS

PostCSS es una herramienta para transformar estilos con plugins JS. Estos plugins pueden limpiar tu CSS, soportar variables y mixins, transpilar sintaxis CSS futura, imágenes en línea, y más.

Visita los siguientes recursos para obtener más información:

- [Sitio web oficial](#)

Herramientas de construcción

Los ejecutores de tareas ejecutan automáticamente comandos y llevan a cabo procesos entre bastidores. Esto ayuda a automatizar su flujo de trabajo mediante la realización de tareas mundanas y repetitivas que de otro modo perdería una cantidad atroz de tiempo repitiendo usted mismo.

Entre los usos comunes de los ejecutores de tareas se incluyen numerosas tareas de desarrollo como: poner en marcha servidores de desarrollo, compilar código (por ejemplo, SCSS a CSS), ejecutar linters, servir archivos desde un puerto local de tu ordenador, ¡y muchas más!

Visita los siguientes recursos para obtener más información:

- [webpack es un agrupador de módulos estáticos para aplicaciones JavaScript modernas](#)
- [Vite, la nueva generación de herramientas frontend](#)
- [Parcel es una herramienta de configuración cero para la web](#)

Agrupadores de módulos

Un agrupador de módulos es una herramienta que toma piezas de JavaScript y sus dependencias y las agrupa en un único archivo, normalmente para su uso en el navegador. Puede que hayas utilizado herramientas como Browserify, Webpack, Rollup o alguna de las muchas otras.

Suele comenzar con un archivo de entrada y, a partir de ahí, agrupa todo el código necesario para ese archivo de entrada.

Visita los siguientes recursos para aprender más:

- [Veamos cómo funcionan los paquetes de módulos](#)
- [Explicación de los paquetes de módulos](#)

Vite

Vite es una herramienta de compilación cuyo objetivo es proporcionar una experiencia de desarrollo más rápida y sencilla para los proyectos web modernos.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Vite](#)
- [Documentación de Vite](#)
- [Curso acelerado de Vite](#)
- [Vídeo tutorial de Vite](#)

esbuild

Nuestras actuales herramientas de compilación para la web son entre 10 y 100 veces más lentas de lo que podrían ser. El objetivo principal del proyecto esbuild bundler es lograr una nueva era en el rendimiento de las herramientas de compilación y, de paso, crear un bundler moderno y fácil de usar.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial de esbuild](#)
- [Documentación de esbuild](#)
- [¿Por qué la gente está obsesionada con esbuild?](#)
- [¿Qué es esbuild?](#)

Webpack

Webpack es un empaquetador de módulos. Su propósito principal es empaquetar archivos JavaScript para su uso en un navegador, pero también es capaz de transformar, empaquetar o empaquetar casi cualquier recurso o activo.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial de Webpack](#)
- [Documentación de Webpack](#)
- [Guía completa de Webpack 5](#)

Rollup

Rollup es un agrupador de módulos para JavaScript que compila pequeños fragmentos de código en algo más grande y complejo, como una biblioteca o una aplicación.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial y documentación](#)
- [Cómo configurar la agrupación de JavaScript mediante Rollup](#)

Parcel

Parcel es un agrupador de aplicaciones web que se distingue por su experiencia para desarrolladores. Ofrece un rendimiento ultrarrápido utilizando procesamiento multinúcleo y no requiere ninguna configuración.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial y documentación](#)
- [Uso de Parcel Bundler con React](#)

Ejecutores de tareas

Los ejecutores de tareas son herramientas que simplifican ciertas tareas tediosas del desarrollo, como la automatización de la compilación sass/scss, la agrupación de activos, el análisis del código fuente y la recarga en caliente del servidor local.

Visite los siguientes recursos para obtener más información:

- [script npm](#)
- [script yarn](#)

Scripts npm

Los scripts npm son las entradas en el campo scripts del archivo package.json. El campo scripts contiene un objeto donde puedes especificar varios comandos y scripts que quieres exponer.

Visite los siguientes recursos para obtener más información:

- [Introducción a los scripts npm](#)
- [Codevolution: scripts npm](#)

Formateadores de linters

Un linter es una herramienta utilizada para analizar código y descubrir fallos, errores sintácticos, incoherencias estilísticas y construcciones sospechosas. Entre los linters más conocidos para JavaScript se encuentran ESLint, JSLint y JSHint.

Visite los siguientes recursos para obtener más información:

- [¿Qué es un linter?](#)

Prettier

Prettier es un formateador de código con soporte para JavaScript, HTML, CSS, YAML, Markdown, GraphQL Schemas. La mayor razón para adoptar Prettier es poner fin a los debates sobre estilos.

Visite los siguientes recursos para saber más:

- [Sitio web de Prettier](#)
- [Por qué Prettier](#)

ESLint

Con ESLint puede imponer el estándar de codificación utilizando un determinado conjunto de reglas independientes.

Visite los siguientes recursos para obtener más información:

- [Página oficial de ESLint](#)
- [Introducción a ESLint](#)
- [ESLint Quickstart - encontrar errores automáticamente](#)

Prueba de aplicaciones

Antes de entregar tu aplicación a los usuarios, tienes que asegurarte de que cumple los requisitos para los que fue diseñada y de que no hace cosas raras e involuntarias (los llamados "bugs"). Para conseguirlo, "probamos" nuestras aplicaciones de distintas maneras.

Visita los siguientes recursos para saber más:

- [Una inmersión completa en las pruebas de software.](#)
- [Los diferentes tipos de pruebas de software](#)
- [Pruebas de aplicaciones React con Jest](#)

Vitest

Vitest es un rápido framework de pruebas unitarias Vite-nativo con soporte out-of-box ESM, TypeScript y JSX. Funciona en React, Vue, Svelte y más proyectos creados con Vite.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Vitest](#)

Jest

Jest es un delicioso marco de pruebas de JavaScript con un enfoque en la simplicidad. Funciona con proyectos que utilizan: Babel, TypeScript, Node, React, Angular, Vue ¡y más!

Visita los siguientes recursos para aprender más:

- [Sitio web de Jest](#)

Playwright

Playwright es una biblioteca de automatización de pruebas de código abierto desarrollada inicialmente por colaboradores de Microsoft. Es compatible con lenguajes de programación como Java, Python, C# y NodeJS. Playwright viene con Licencia Apache 2.0 y es más popular con NodeJS con Javascript/Typescript.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Playwright](#)
- [Tutorial de Playwright: Aprenda lo básico y la configuración](#)
- [Playwright, un marco de pruebas integral que ahorra tiempo](#)

Cypress

Cypress framework es un marco de pruebas de extremo a extremo basado en JavaScript construido sobre Mocha - un marco de pruebas JavaScript rico en características que se ejecuta sobre y en el navegador, haciendo que las pruebas asíncronas sean simples y convenientes. También utiliza una biblioteca de aserción BDD/TDD y un navegador para emparejarse con cualquier marco de pruebas JavaScript.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentación oficial](#)
- [Pruebas integrales de Cypress](#)

Estrategias de autenticación

Las estrategias de autenticación son métodos o técnicas utilizados para verificar la identidad de un usuario o sistema con el fin de conceder acceso a un recurso protegido. Existen diferentes estrategias de autenticación que pueden utilizarse, entre las que se incluyen:

- Autenticación básica
- Autenticación basada en sesión
- Autenticación basada en token
- Autenticación JWT
- OAuth
- SSO

No tienes por qué aprenderlas todas, ni cómo ponerlas en práctica, ni los entresijos desde el principio. Pero es importante saber cuáles son y cómo funcionan. Esto te ayudará a tomar mejores decisiones a la hora de elegir una estrategia de autenticación para tu aplicación.

Visita los siguientes recursos para obtener más información:

- [Autenticación básica](#)
- [Autenticación basada en sesión](#)
- [Autenticación basada en token](#)
- [Autenticación JWT](#)
- [OAuth](#)
- [SSO - Inicio de sesión único](#)

Conocimientos de seguridad web

La seguridad web se refiere a las medidas de protección tomadas por los desarrolladores para proteger las aplicaciones web de amenazas que podrían afectar al negocio.

Visite los siguientes recursos para obtener más información:

- [Por qué es importante HTTPS](#)
- [Wikipedia - OWASP](#)
- [Lista de comprobación de seguridad de aplicaciones web OWASP](#)
- [Los 10 principales riesgos de seguridad OWASP](#)
- [OWASP Cheatsheet](#)
- [Política de seguridad de contenidos \(CSP\)](#)
- [Tutorial OWASP ZAP paso a paso](#)

CORS

Cross-Origin Resource Sharing (CORS) es un mecanismo basado en cabeceras HTTP que permite a un servidor indicar cualquier origen (dominio, esquema o puerto) distinto del suyo desde el que un navegador debe permitir la carga de recursos.

Visite los siguientes recursos para obtener más información:

- [CORS - Intercambio de recursos entre orígenes](#)

HTTPS

HTTPS es una forma segura de enviar datos entre un servidor web y un navegador.

El protocolo seguro de transferencia de hipertexto (HTTPS) es la versión segura de HTTP, que es el protocolo principal utilizado para enviar datos entre un navegador web y un sitio web. HTTPS está cifrado para aumentar la seguridad de la transferencia de datos. Esto es especialmente importante cuando los usuarios transmiten datos confidenciales, como al iniciar sesión en una cuenta bancaria, un servicio de correo electrónico o un proveedor de seguro médico.

Visite los siguientes recursos para obtener más información:

- [¿Qué es HTTPS?](#)
- [Por qué es importante HTTPS](#)
- [Activar HTTPS en sus servidores](#)
- [Cómo funciona HTTPS \(cómic\)](#)
- [Explicación de SSL, TLS, HTTP y HTTPS](#)
- [HTTPS - Historias desde el terreno](#)

Política de seguridad de contenidos

La Política de Seguridad de Contenidos es un estándar de seguridad informática introducido para prevenir ataques de cross-site scripting, clickjacking y otros ataques de inyección de código resultantes de la ejecución de contenido malicioso en el contexto de una página web de confianza.

Visite los siguientes recursos para obtener más información:

- [Política de seguridad de contenidos \(CSP\) de MDN](#)
- [Política de seguridad de contenidos \(CSP\) de Google Devs](#)
- [Web.dev - Política de seguridad de contenidos \(CSP\)](#)

Riesgos de seguridad OWASP

OWASP u Open Web Application Security Project es una comunidad en línea que produce artículos, metodologías, documentación, herramientas y tecnologías de libre acceso en el campo de la seguridad de las aplicaciones web.

Visite los siguientes recursos para obtener más información:

- [Wikipedia - OWASP](#)
- [Lista de comprobación de seguridad de aplicaciones web OWASP](#)
- [Los 10 mayores riesgos de seguridad OWASP](#)
- [OWASP Cheatsheet](#)
- [OWASP Top 10: Guía de las peores vulnerabilidades de software](#)

Web Components

Web Components es un conjunto de diferentes tecnologías que le permiten crear elementos personalizados reutilizables - con su funcionalidad encapsulada lejos del resto de su código - y utilizarlos en sus aplicaciones web.

Visite los siguientes recursos para obtener más información:

- [Web Components | MDN](#)
- [WebComponents.org](#)
- [Curso intensivo de componentes web](#)

Plantillas HTML

El elemento HTML `<template>` es un mecanismo para almacenar HTML que no se va a mostrar inmediatamente cuando se carga una página, sino que se puede instanciar posteriormente durante el tiempo de ejecución utilizando JavaScript. Piense en una plantilla como un fragmento de contenido que se almacena para su uso posterior en el documento.

Visite los siguientes recursos para obtener más información:

- [Uso de plantillas y ranuras | MDN web docs](#)

Custom Elements

Una de las características clave del estándar Web Components es la posibilidad de crear elementos personalizados que encapsulan su funcionalidad en una página HTML, en lugar de tener que conformarse con un largo lote anidado de elementos que, juntos, proporcionan una característica de página personalizada.

Visite los siguientes recursos para obtener más información:

- [Uso de Custom Elements | MDN web docs](#)

Shadow DOM

Un aspecto importante de los componentes web es la encapsulación: ser capaz de mantener la estructura de marcado, el estilo y el comportamiento ocultos y separados del resto del código de la página, de modo que las distintas partes no choquen entre sí y el código pueda mantenerse limpio y ordenado. La API Shadow DOM es una parte clave de esto, ya que proporciona una forma de adjuntar un DOM separado y oculto a un elemento.

Visita los siguientes recursos para obtener más información:

- [Uso de shadow DOM | MDN web docs](#)

Comprobadores de tipos

Los comprobadores de tipos ayudan a los desarrolladores a escribir código con menos errores añadiendo tipos a su código, intentando detectar errores de tipo dentro de su código y eliminándolos después en tiempo de compilación. Flow y TypeScript son dos populares comprobadores estáticos de tipos para JavaScript.

TypeScript

TypeScript es un lenguaje de programación fuertemente tipado que se basa en JavaScript, ofreciéndole mejores herramientas a cualquier escala.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentos oficiales para inmersiones profundas](#)
- [Manual de TypeScript](#)
- [Tutorial de TypeScript](#)
- [Scrimba - Conceptos básicos de TypeScript](#)
- [TypeScript para principiantes](#)
- [¿Qué es TypeScript?](#)
- [Tutorial de TypeScript: Más allá del "¡Hola, mundo!"](#)
- [Tutorial de TypeScript: Guía de uso del lenguaje de programación](#)

Renderizado del lado del servidor

La renderización del lado del servidor se refiere al proceso por el que el lado del servicio completa el empalme de la estructura HTML de la página, la envía al navegador y, a continuación, vincula el estado y los eventos para que se convierta en una página totalmente interactiva.

Visite los siguientes recursos para obtener más información:

- [Qué es el renderizado del lado del servidor](#)
- [¿Cuándo debo renderizar del lado del servidor?](#)
- [Renderizado del lado del servidor \(SSR\)](#)
- [¿Qué es el renderizado del lado del servidor?](#)
- [¿Qué es el renderizado del lado del servidor para el desarrollo web?](#)

React

React es la biblioteca JavaScript más popular para la creación de interfaces de usuario. React también puede renderizar en el servidor utilizando Node y potenciar aplicaciones móviles utilizando React Native.

Visite los siguientes recursos para obtener más información:

- [Visite la hoja de ruta dedicada de React](#)
- [Sitio web de React](#)
- [Introducción oficial](#)
- [Guía de React para principiantes](#)
- [Curso de React JS para principiantes](#)
- [Curso de React - Tutorial para principiantes sobre la biblioteca JavaScript de React \[2022\]](#)
- [Entendiendo el Proceso de Renderizado de la UI de React](#)

Next.js

Next.js es un marco de desarrollo de código abierto construido sobre Node.js que permite funcionalidades de aplicaciones web basadas en React como la renderización del lado del servidor y la generación de sitios web estáticos.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentos oficiales para empezar](#)
- [Dominar Next.js](#)
- [Manual de Next.js - freeCodeCamp](#)
- [Next.js para principiantes – freeCodeCamp](#)

Remix

Remix es un framework web de pila completa que te permite centrarte en la interfaz de usuario y retroceder a través de los estándares web para ofrecer una experiencia de usuario rápida, hábil y resistente. A la gente le encantará usar tus cosas.

Visita los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentos oficiales para empezar](#)

Angular

Angular es un marco de desarrollo front-end basado en componentes construido sobre TypeScript que incluye una colección de librerías bien integradas que incluyen características como enrutamiento, gestión de formularios, comunicación cliente-servidor y mucho más.

Visite los siguientes recursos para obtener más información:

- [Oficial - Primeros pasos con Angular](#)

Angular Universal

El proyecto Angular Universal es un proyecto impulsado por la comunidad para ampliar las API del núcleo de Angular (plataforma-servidor) para permitir a los desarrolladores hacer el renderizado del lado del servidor de las aplicaciones Angular. Utiliza principalmente express para renderizar páginas en páginas en el servidor node.js.

Visita los siguientes recursos para aprender más:

- [Renderizado del lado del servidor \(SSR\) con Angular Universal](#)

Vue.js

Vue.js es un framework JavaScript de código abierto para crear interfaces de usuario y aplicaciones de una sola página. Se centra principalmente en el desarrollo front-end.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Vue.js](#)
- [Introducción oficial](#)
- [Curso de Vue.js para principiantes](#)
- [Curso acelerado de Vue.js](#)
- [Conozca Vue.js, el framework JavaScript flexible](#)

Nuxt.js

Nuxt.js es una librería JavaScript gratuita y de código abierto basada en Vue.js, Node.js, Webpack y Babel.js. Nuxt se inspira en Next.js, que es un framework de propósito similar, basado en React.js.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Nuxt](#)
- [Documentos Nuxt](#)
- [Tutoriales Nuxt](#)
- [Introducción a Nuxt](#)
- [Fundamentos de Nuxt.js](#)

Svelte

Svelte es un framework javascript que a diferencia de Vue y React no utiliza DOM diffing vertical sino que sabe exactamente qué y dónde actualizar cuando el estado cambia. Se centra principalmente en frontend y la construcción de interfaces de usuario.

Visita los siguientes recursos para aprender más:

- [Sitio web de Svelte](#)
- [Documentación de Svelte](#)
- [Lista de reproducción del curso de Svelte para principiantes](#)
- [Todo sobre Svelte, el marco de trabajo web basado en estados más apreciado](#)
- [Svelte y el futuro del desarrollo Frontend](#)

Kit Svelte

Svelte Kit es un framework para construir aplicaciones web con el framework Svelte JavaScript. Está diseñado para ser una solución completa para la creación de aplicaciones web, con soporte para el renderizado del lado del servidor, división automática del código y rendimiento optimizado.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial de Svelte Kit](#)
- [Documentos oficiales del kit Svelte](#)

GraphQL

GraphQL es un lenguaje de consulta para API y un tiempo de ejecución para realizar esas consultas con los datos existentes. GraphQL proporciona una descripción completa y comprensible de los datos de su API, ofrece a los clientes la posibilidad de pedir exactamente lo que necesitan y nada más, facilita la evolución de las API a lo largo del tiempo y habilita potentes herramientas para desarrolladores.

Visite los siguientes recursos para obtener más información:

- [Introducción a GraphQL](#)
- [Tutorial Fullstack para GraphQL](#)
- [Tutoriales de GraphQL](#)
- [Curso de GraphQL para principiantes](#)
- [Introducción a GraphQL](#)
- [Cómo ejecutar una consulta GraphQL sencilla](#)

Apollo

Apollo es una plataforma para construir un gráfico unificado, una capa de comunicación que te ayuda a gestionar el flujo de datos entre los clientes de tu aplicación (como aplicaciones web y nativas) y tus servicios back-end.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Apollo](#)
- [Documentos oficiales](#)
- [Canal oficial de YouTube](#)
- [Tutorial de GraphQL con React - Cliente Apollo](#)

Relay Modern

Relay es un cliente JavaScript utilizado en el navegador para obtener datos GraphQL. Es un framework JavaScript desarrollado por Facebook para gestionar y obtener datos en aplicaciones React. Está construido pensando en la escalabilidad para alimentar aplicaciones complejas como Facebook. El objetivo final de GraphQL y Relay es ofrecer interacciones instantáneas de interfaz de usuario-respuesta.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Introducción al relevo moderno](#)

Generadores de sitios estáticos

Un generador de sitios estáticos es una herramienta que genera un sitio web HTML estático completo a partir de datos sin procesar y un conjunto de plantillas. Esencialmente, un generador de sitios estáticos automatiza la tarea de codificar páginas HTML individuales y las prepara para servirlas a los usuarios con antelación. Como estas páginas HTML están preconstruidas, pueden cargarse muy rápidamente en los navegadores de los usuarios.

Visite los siguientes recursos para obtener más información:

- [¿Qué es un generador de sitios estáticos?](#)
- [Next.js SSG](#)
- [Gatsby SSG](#)
- [SSG - Un sándwich de 11ty, Vite y JAM](#)
- [Volver a lo básico con generadores de sitios web estáticos](#)

Astro

Astro es un framework web todo en uno para construir sitios web rápidos y centrados en el contenido. Astro combina la potencia de un moderno framework basado en componentes con el rendimiento y la flexibilidad de un generador de sitios estáticos.

- Islas de componentes: Una nueva arquitectura web para crear sitios web más rápidos.
- Diseño de API que da prioridad al servidor: Mueva la costosa hidratación fuera de los dispositivos de sus usuarios.
- Cero JS, por defecto: No hay sobrecarga de tiempo de ejecución de JavaScript que te ralentice.
- Preparado para el borde: Despliegue en cualquier lugar, incluso en un tiempo de ejecución de borde global como Deno o Cloudflare.
- Personalizable: Tailwind, MDX y más de 100 integraciones entre las que elegir.
- Interfaz de usuario agnóstica: compatible con React, Preact, Svelte, Vue, Solid, Lit y más.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial de Astro](#)
- [Documentos oficiales de Astro](#)

Eleventy

Eleventy (11ty) es un generador de sitios estáticos sencillo de usar, fácil de personalizar, de alto rendimiento y potente, con un útil conjunto de complementos (por ejemplo, navegación, transformaciones de imágenes en tiempo de compilación, activos de caché). Las páginas pueden construirse y escribirse con una gran variedad de lenguajes de plantilla (HTML, Markdown, JavaScript, Liquid, Nunjucks, Handlebars, Mustache, EJS, Haml, Pug o literales de plantilla JS). Pero también ofrece la posibilidad de crear dinámicamente páginas a partir de datos locales o fuentes externas que se compilan en tiempo de compilación. No tiene dependencias de JavaScript del lado del cliente.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)

- [Una colección de 11ty starters, proyectos, plugins y recursos](#)
- [Introducción a Eleventy](#)

Next.js

Next.js es un marco de desarrollo de código abierto construido sobre Node.js que permite funcionalidades de aplicaciones web basadas en React como la renderización del lado del servidor y la generación de sitios web estáticos.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentos oficiales para empezar](#)
- [Manual de Next.js - FreeCodeCamp](#)
- [Dominar Next.js](#)
- [Next.js para principiantes - Curso completo en el canal de YouTube de freeCodeCamp](#)

Remix

Remix es un framework web de pila completa que te permite centrarte en la interfaz de usuario y retroceder a través de los estándares web para ofrecer una experiencia de usuario rápida, hábil y resistente. A la gente le encantará usar tus cosas.

Visita los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentos oficiales para empezar](#)

Vuepress

VuePress se compone de dos partes: un generador de sitios estáticos minimalista (abre nueva ventana) con un sistema de tematización basado en Vue y una API de plugins, y un tema por defecto optimizado para escribir documentación técnica. Fue creado para apoyar las necesidades de documentación de los propios subproyectos de Vue.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentación oficial para empezar](#)
- [Introducción a VuePress](#)

Jekyll

Jekyll es un generador de sitios estáticos. Toma el texto escrito en su lenguaje de marcado favorito y utiliza diseños para crear un sitio web estático. Puede modificar el aspecto del sitio, las URL, los datos que aparecen en la página y mucho más.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Jekyll](#)

- [Documentos de Jekyll](#)

Hugo

Hugo es el motor de sitios web estáticos más rápido del mundo. Está escrito en Go (también conocido como Golang) y desarrollado por bep, spf13 y sus amigos.

Visita los siguientes recursos para saber más:

- [Sitio web oficial](#)
- [Documentos oficiales para empezar](#)
- [Introducción a Hugo - Generador de sitios estáticos](#)
- [Tutorial: Usar Hugo para generar un sitio web estático](#)

Nuxt.js

Nuxt.js es un marco de desarrollo de código abierto construido sobre Node.js que permite funcionalidades de aplicaciones web basadas en Vue como la renderización del lado del servidor y la generación de sitios web estáticos.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentación oficial para empezar](#)
- [Dominio de Nuxt.js](#)
- [Vue.js para principiantes - Curso completo en el canal de YouTube de freeCodeCamp](#)

Aplicaciones web progresivas

Las aplicaciones web progresivas (PWA) son sitios web que se mejoran progresivamente para funcionar como aplicaciones nativas instaladas en plataformas compatibles, al tiempo que funcionan como sitios web normales en otros navegadores.

Visite los siguientes recursos para obtener más información:

- [Aplicaciones web progresivas para principiantes](#)
- [Aprender PWA](#)
- [MDN Web Docs: Aplicaciones Web Progresivas](#)
- [Crear una aplicación web progresiva](#)

Calcular, medir y mejorar el rendimiento

Patrón PRPL

El patrón PRPL es un patrón de optimización del rendimiento para aplicaciones web que se centra en minimizar el tiempo que tarda el renderizado inicial de una página web. Sus siglas significan Push, Render, Pre-cache, y Lazy-load, e implica los siguientes pasos:

- Push: Dar prioridad a la entrega de recursos críticos, como HTML, CSS y JavaScript, al cliente lo antes posible.
- Renderizar: Empezar a renderizar la página web tan pronto como se reciban los recursos críticos, incluso si todavía se están descargando algunos recursos no críticos.
- Precache: Almacena en caché los recursos no críticos en segundo plano para que estén disponibles cuando se necesiten.
- Carga diferida: Aplazar la carga de recursos no críticos hasta que se necesiten, como cuando el usuario se desplaza hasta ellos o interactúa con ellos.

El patrón PRPL está diseñado para mejorar el rendimiento percibido de una página web reduciendo el tiempo que tarda la página en volverse interactiva. Es especialmente útil para aplicaciones que se sirven a través de redes lentas o poco fiables, ya que permite que la página se renderice lo más rápidamente posible y luego cargue gradualmente los recursos restantes.

Visite los siguientes recursos para obtener más información:

- [Patrón PRPL - Google Developers](#)

Modelo RAIL

El modelo RAIL es un modelo de optimización del rendimiento para aplicaciones web que se centra en mejorar el rendimiento percibido de una página web. Son las siglas de Response (respuesta), Animation (animación), Idle (inactividad) y Load (carga), e implica los siguientes cuatro objetivos de rendimiento:

- Respuesta: El tiempo que tarda la página web en responder a la entrada del usuario debe ser inferior a 100 milisegundos.
- Animación: El tiempo que tarda una animación en ejecutarse debe ser inferior a 10 milisegundos.

- **Ralentí:** La página web debe aprovechar el tiempo de inactividad para realizar tareas no críticas.
- **Carga:** el tiempo que tarda la página web en cargarse completamente debe ser inferior a 1 segundo.

El modelo RAIL está diseñado para proporcionar un marco que permita optimizar el rendimiento de una página web centrándose en las cuatro áreas clave que influyen en la percepción del rendimiento por parte del usuario. Es especialmente útil para aplicaciones que requieren altos niveles de interactividad, como juegos y aplicaciones de redes sociales.

Visite los siguientes recursos para obtener más información:

- [Modelo RAIL - Google Developers](#)

Métricas de rendimiento

Las métricas de rendimiento web son medidas cuantitativas del rendimiento de una página o aplicación web. Se utilizan para evaluar la velocidad y eficacia de una página web, y pueden ayudar a identificar áreas de mejora. Algunas métricas comunes de rendimiento web incluyen:

- **Tiempo de carga:** El tiempo que tarda una página web en cargarse completamente y volverse interactiva.
- **Primera pintura de contenido (FCP):** El tiempo que tarda en aparecer el primer contenido en la página.
- **Tiempo hasta la interactividad (TTI):** El tiempo que tarda la página en ser totalmente interactiva.
- **Retraso de la primera entrada (FID):** El tiempo que tarda la página en responder a la primera entrada del usuario.
- **Tiempo total de bloqueo (TBT):** El tiempo que tarda la página en ser totalmente interactiva, teniendo en cuenta el tiempo de bloqueo del hilo principal.

Hay muchas herramientas y técnicas disponibles para medir las métricas de rendimiento web, incluidas las herramientas de desarrollo del navegador, las herramientas de supervisión del rendimiento y las API de rendimiento web. Mediante el seguimiento de estas métricas y el análisis de los resultados, los desarrolladores web pueden identificar áreas de mejora y optimizar el rendimiento de sus páginas web.

Visite los siguientes recursos para obtener más información:

- [Métricas de rendimiento web - Google Developers](#)
- [Medición del rendimiento y de la experiencia del usuario - Google Developers](#)
- [Medir el rendimiento y la experiencia del usuario – MDN](#)

Lighthouse

Lighthouse es una herramienta de código abierto desarrollada por Google que se utiliza para auditar el rendimiento, la accesibilidad y el SEO de las páginas web. Está disponible como extensión del navegador y como herramienta de línea de comandos, y puede ejecutarse en cualquier página web para generar un informe con recomendaciones de mejora.

Lighthouse funciona simulando la carga y la interacción de una página web y midiendo varias métricas de rendimiento, como el tiempo de carga, el tiempo hasta la primera pintura y el tiempo

hasta la interactividad. También comprueba problemas comunes como el tamaño incorrecto de las imágenes, la falta de texto alternativo y los enlaces rotos.

Lighthouse proporciona una herramienta completa y fácil de usar para identificar y solucionar problemas de rendimiento y accesibilidad en páginas web. Es muy utilizada por los desarrolladores web y está integrada en muchas herramientas de desarrollo populares.

Visite los siguientes recursos para obtener más información:

- [Lighthouse - Desarrolladores de Google](#)
- [Mejora del rendimiento de carga - Chrome DevTools 101](#)

Herramientas de desarrollo del navegador

La mayoría de los navegadores tienen herramientas de desarrollo integradas que te ayudan a realizar tareas específicas de desarrollo web en una página web. Mi recomendación personal sería utilizar Chrome DevTools, ya que es el más avanzado y tiene más funciones. Sin embargo, si estás utilizando un navegador diferente, puedes seguir utilizando las DevTools para depurar tu aplicación web.

Chrome DevTools es un conjunto de herramientas de desarrollo web integradas en el navegador Google Chrome. Permite a los desarrolladores web depurar y optimizar páginas web proporcionando una serie de funciones para inspeccionar y manipular el HTML, CSS y JavaScript de la página.

Algunas de las funciones que ofrece Chrome DevTools son:

- **Panel de elementos:** Inspecciona y modifica el HTML y CSS de la página.
- **Panel de consola:** Ver y depurar errores de JavaScript y mensajes de registro.
- **Panel de red:** Supervisa las solicitudes y respuestas de la red y analice los problemas de rendimiento.
- **Panel de rendimiento:** Analice el rendimiento de la página e identifique cuellos de botella.
- **Panel de aplicaciones:** Inspeccione los recursos de la página, como las cookies y el almacenamiento local.

Chrome DevTools es una herramienta potente y esencial para los desarrolladores web, y se utiliza ampliamente para depurar y optimizar páginas web. Se actualiza constantemente con nuevas funciones y mejoras, y está disponible en todos los navegadores web modernos.

Visita los siguientes recursos para obtener más información:

- [Chrome DevTools - Google Developers](#)

Conozca las diferentes API web utilizadas en las PWA

API de almacenamiento web

La API de almacenamiento web proporciona mecanismos para almacenar pares clave-valor en un navegador web. Incluye dos objetos de almacenamiento: `localStorage` y `sessionStorage`, que permiten guardar datos en el lado del cliente y conservarlos en varias sesiones del navegador, respectivamente.

La API de almacenamiento web está diseñada para ser sencilla y fácil de usar, y es ampliamente compatible con los navegadores web modernos. A menudo se utiliza como alternativa a las cookies, ya que permite almacenar mayores cantidades de datos y es más eficiente en términos de rendimiento.

Visite los siguientes recursos para obtener más información:

- [API de almacenamiento web – MDN](#)

Web Sockets

Web Sockets es una tecnología que permite la comunicación full-duplex a través de una única conexión TCP. Permite la comunicación bidireccional en tiempo real entre un cliente y un servidor, y se utiliza normalmente en aplicaciones que requieren una comunicación de alta velocidad y baja latencia, como los juegos en línea y la transmisión de datos en tiempo real.

Web Sockets utiliza una conexión persistente entre un cliente y un servidor, lo que permite un intercambio continuo de datos sin necesidad de que el cliente envíe peticiones adicionales al servidor. Esto la hace más eficiente y rápida que otras tecnologías, como HTTP, que requieren el envío de una nueva solicitud para cada dato.

Web Sockets es compatible con la mayoría de los navegadores web modernos y puede utilizarse con diversos lenguajes y marcos de programación.

Visite los siguientes recursos para obtener más información:

- [Web Sockets – MDN](#)

Eventos enviados por el servidor

Server-Sent Events (SSE) es una tecnología que permite a un servidor web enviar datos a un cliente en tiempo real. Utiliza una conexión HTTP para enviar un flujo de datos del servidor al cliente, y el cliente puede escuchar estos eventos y actuar cuando los recibe.

SSE es útil para aplicaciones que requieren actualizaciones en tiempo real, como sistemas de chat, teletipos de bolsa y feeds de redes sociales. Es una forma sencilla y eficaz de establecer una conexión duradera entre un cliente y un servidor, y es compatible con la mayoría de los navegadores web modernos.

Para utilizar SSE, el cliente debe crear un objeto EventSource y especificar la URL del script del servidor que enviará los eventos. A continuación, el servidor puede enviar los eventos escribiéndolos en el flujo de respuesta con el formato adecuado.

Visite los siguientes recursos para obtener más información:

- [Eventos enviados por el servidor – MDN](#)

Service Workers

Los Service Workers son un tipo de web worker que actúa como proxy entre una página web y la red, permitiendo a los desarrolladores web construir aplicaciones offline-first y fiables. Los Service Workers pueden interceptar peticiones de red, acceder a la caché y tomar decisiones sobre cómo responder a una petición en función de los recursos disponibles.

Los Service Workers están escritos en JavaScript y son registrados por una página web. Una vez registrados, pueden controlar la página y todas sus peticiones, incluso cuando la página no está abierta en un navegador. Esto permite a los Service Workers habilitar funciones como las notificaciones push, la sincronización en segundo plano y el soporte offline.

Los Service Workers son compatibles con la mayoría de los navegadores web modernos y son un componente esencial de las aplicaciones web progresivas (PWA).

Visita los siguientes recursos para obtener más información:

- [Service Workers – MDN](#)

API de geolocalización

La API de geolocalización es una API web que proporciona acceso a los datos de localización del dispositivo, como la latitud y la longitud. Permite a los desarrolladores web crear aplicaciones basadas en la ubicación, como mapas y localización compartida, utilizando el GPS, Wi-Fi y otros sensores del dispositivo para determinar la ubicación del usuario.

Para utilizar la API de geolocalización, una página web debe pedir permiso al usuario para acceder a su ubicación. Si se concede el permiso, la página puede utilizar el objeto `navigator.geolocation` para acceder a los datos de localización del dispositivo. La API proporciona varios métodos para obtener la ubicación actual del usuario, vigilar los cambios de ubicación y calcular las distancias entre dos ubicaciones.

Visite los siguientes recursos para obtener más información:

- [API de geolocalización – MDN](#)

API de notificaciones

La API de notificaciones es una API web que permite a las páginas web mostrar notificaciones del sistema al usuario. Estas notificaciones pueden utilizarse para alertar al usuario de eventos importantes, como nuevos mensajes o actualizaciones, incluso cuando la página web no está abierta en el navegador.

Para utilizar la API de notificaciones, una página web debe solicitar primero permiso al usuario para mostrar notificaciones. Si se concede el permiso, la página puede utilizar el constructor de `Notification` para crear una nueva notificación y mostrarla al usuario. La notificación puede incluir un título, un cuerpo de texto y un icono, y puede personalizarse con opciones como un tiempo de espera y una acción de clic.

Visite los siguientes recursos para obtener más información:

- [API de notificaciones - MDN](#)
- [Crear notificaciones React con la API de notificaciones web](#)

API de orientación del dispositivo

La API de orientación del dispositivo es una API web que proporciona acceso a los datos de orientación y movimiento del dispositivo, como su cabeceo, balanceo y guiñada. Permite a los desarrolladores web crear aplicaciones que pueden responder a la orientación y el movimiento del dispositivo, como la realidad aumentada y los juegos controlados por movimiento.

Para utilizar la API de orientación del dispositivo, una página web debe pedir permiso al usuario para acceder a los datos de orientación del dispositivo. Si se concede el permiso, la página puede utilizar el objeto `DeviceOrientationEvent` para acceder a los datos de orientación del dispositivo y responder a los cambios de orientación. La API proporciona varias propiedades para acceder a los datos de orientación y movimiento del dispositivo, así como eventos para detectar cambios en la orientación.

La API de orientación del dispositivo es compatible con la mayoría de los navegadores web modernos y suele utilizarse junto con otras API, como la API de geolocalización, para crear aplicaciones basadas en la ubicación.

Visite los siguientes recursos para obtener más información:

- [API de orientación de dispositivos – MDN](#)

Pagos

La API de solicitud de pago es una API web que permite a los desarrolladores crear flujos de pago en sus aplicaciones web. Proporciona una interfaz estandarizada basada en navegador para recopilar información de pago y envío del usuario, y admite una amplia gama de métodos de pago, incluidas tarjetas de crédito, tarjetas de débito y monederos digitales.

Para utilizar la API de solicitud de pago, una página web debe crear primero un objeto `PaymentRequest` y especificar las opciones de pago y envío disponibles para el usuario. A continuación, la página puede invocar la interfaz de usuario de solicitud de pago llamando al método `show()` del objeto `PaymentRequest`. A continuación, el usuario puede seleccionar sus opciones de pago y envío preferidas y confirmar el pago, momento en el que la API de solicitud de pago devolverá un objeto de respuesta de pago que puede utilizarse para completar la transacción.

La API de solicitud de pago es compatible con la mayoría de los navegadores web modernos y está diseñada para ser sencilla y eficaz tanto para el desarrollador como para el usuario.

Visite los siguientes recursos para obtener más información:

- [API de solicitud de pago – MDN](#)

API de credenciales

La API de gestión de credenciales es una API web que permite a los desarrolladores web integrar flujos de inicio de sesión federados y basados en contraseñas en sus aplicaciones. Proporciona una interfaz estandarizada basada en navegador para almacenar y recuperar credenciales de usuario, como combinaciones de nombre de usuario y contraseña y tokens OAuth.

Para utilizar la API de gestión de credenciales, una página web debe crear primero un objeto `Credential` y especificar las credenciales que el usuario desea almacenar. A continuación, la página puede utilizar el objeto `navigator.credentials` para almacenar y recuperar las credenciales del usuario. La API proporciona varios métodos para almacenar y recuperar credenciales, así como para solicitar al usuario que inicie sesión o se registre.

La API de gestión de credenciales es compatible con la mayoría de los navegadores web modernos y está diseñada para mejorar la seguridad y la facilidad de uso de los flujos de inicio de sesión al permitir al usuario almacenar y reutilizar sus credenciales en varios sitios y dispositivos.

Visite los siguientes recursos para obtener más información:

- [API de gestión de credenciales – MDN](#)

Aplicaciones móviles

Hace un tiempo, desarrollar una aplicación móvil con JavaScript era imposible. Pero ahora los desarrolladores de JavaScript pueden crear aplicaciones móviles utilizando sus conocimientos para el desarrollo web. Aquí está la lista de opciones para crear aplicaciones móviles en JavaScript.

- [React Native](#)
- [Flutter](#)
- [Ionic](#)
- [NativeScript](#)

React Native

React Native es un popular marco de aplicaciones móviles basado en JavaScript que permite crear aplicaciones móviles con renderizado nativo para iOS y Android. El framework te permite crear una aplicación para varias plataformas utilizando el mismo código base.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Introducción oficial a React Native](#)
- [Crear una aplicación React Native por Mosh](#)
- [Aprender React Native por CodeAcademy](#)

Flutter

Flutter es un framework de interfaz de usuario móvil gratuito y de código abierto creado por Google y lanzado en mayo de 2017. En pocas palabras, permite crear una aplicación móvil nativa con una sola base de código. Esto significa que puedes utilizar un lenguaje de programación y una base de código para crear dos apps diferentes (para iOS y Android).

Flutter consta de dos partes importantes:

- Un SDK (Kit de Desarrollo de Software): Una colección de herramientas que te van a ayudar a desarrollar tus aplicaciones. Esto incluye herramientas para compilar tu código en código máquina nativo (código para iOS y Android).
- Un Framework (Biblioteca de UI basada en widgets): Una colección de elementos de interfaz de usuario reutilizables (botones, entradas de texto, deslizadores, etc.) que puedes personalizar para tus propias necesidades. Para desarrollar con Flutter, utilizarás un lenguaje de programación llamado Dart. El lenguaje fue creado por Google en octubre de 2011, pero ha mejorado mucho en estos últimos años.

Dart se centra en el desarrollo front-end, y puedes utilizarlo para crear aplicaciones móviles y web.

Si sabes un poco de programación, Dart es un lenguaje de programación de objetos tipados. Puedes comparar la sintaxis de Dart con la de JavaScript.

Visite los siguientes recursos para obtener más información:

- [Visite la hoja de ruta dedicada de Flutter](#)
- [Sitio web de Flutter](#)

- [Tutorial de Flutter](#)
- [Tutorial de Flutter para principiantes](#)
- [Aprende a programar con Dart](#)
- [12 formas en que Flutter agiliza el desarrollo de aplicaciones](#)

Ionic

Ionic framework es un conjunto de herramientas de interfaz de usuario de código abierto para crear aplicaciones móviles, aplicaciones de escritorio y aplicaciones web progresivas eficaces y de alta calidad utilizando tecnologías web como HTML, CSS y JavaScript.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Introducción oficial a Ionic Framework](#)
- [Ionic Framework 4 - Tutorial completo](#)

NativeScript

NativeScript es un framework de código abierto para crear aplicaciones nativas para iOS y Android en Angular, TypeScript o JavaScript.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Introducción oficial a NativeScript](#)
- [Tutorial de NativeScript para principiantes](#)

Aplicaciones de escritorio en JavaScript

Hace un tiempo, desarrollar una aplicación de escritorio utilizando JavaScript era imposible. Pero ahora los desarrolladores de JavaScript pueden crear aplicaciones de escritorio utilizando sus conocimientos para el desarrollo web. Aquí está la lista de opciones para crear aplicaciones de escritorio en JavaScript.

- [Electron](#)
- [NodeGUI](#)
- [NW.js](#)
- [Meteor](#)

Electron

Electron permite crear aplicaciones de escritorio multiplataforma con HTML, CSS y Javascript/Typescript. Utiliza Chromium y Node.js, por lo que esencialmente es una aplicación similar a un "Navegador" que es compatible con Mac, Windows y Linux.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Electron](#)
- [Documentos de Electron](#)
- [Crear una aplicación de escritorio con JavaScript y Electron](#)

Tauri

Tauri es un conjunto de herramientas que ayuda a los desarrolladores a crear aplicaciones para las principales plataformas de escritorio, utilizando prácticamente cualquier framework frontend existente. El núcleo está construido con Rust, y la CLI aprovecha Node.js haciendo de Tauri un enfoque genuinamente políglo to para crear y mantener grandes aplicaciones.

Visita los siguientes recursos para obtener más información:

- [Sitio web de Tauri](#)
- [Documentación de Tauri](#)
- [Cómo Tauri convierte diseños web en aplicaciones nativas compactas](#)

Flutter

Flutter es un framework de interfaz de usuario multiplataforma gratuito y de código abierto creado por Google y lanzado en mayo de 2017. En pocas palabras, permite crear una aplicación móvil nativa con una sola base de código. Esto significa que puedes utilizar un lenguaje de programación y una base de código para crear tres apps diferentes (para iOS, Android y escritorio).

Visite los siguientes recursos para obtener más información:

- [Visita la hoja de ruta dedicada de Flutter](#)
- [Sitio web de Flutter](#)

- [Flutter para escritorio](#)
- [Tutorial de Flutter](#)
- [Tutorial de Flutter para principiantes](#)
- [Aprende a programar con Dart](#)
- [12 formas en que Flutter agiliza el desarrollo de aplicaciones](#)

Contenido adicional

No es necesario completar el contenido adicional, sólo amplía los temas tratados en la hoja de ruta y contiene los elementos que se omitieron en la hoja de ruta. Repáselos si lo desea.

- **UI / UX Conocimiento** - La obtención de UI / UX conocimiento no es una necesidad para un desarrollador Frontend, pero tener un ojo para el diseño y una buena comprensión de UI / UX le ayudará en su carrera.
- **Sistemas de diseño** - Aprende a crear y trabajar con sistemas de diseño.
- **Programación Visual** - es totalmente posible ser un desarrollador frontend competente y exitoso sin ella, pero para muchos desarrolladores frontend este camino representa una parte importante de su carga de trabajo, y representa una opción muy demandada para el desarrollo frontend - especialmente para los desarrolladores que tienen una aptitud visual.
 - Para animaciones: **GSAP**, **Lottie**, **Framer Motion** o **MoJs**
 - Biblioteca de gráficos web (WebGL): **WebGL MDN**, **Fundamentos de WebGL**
 - Gráficos 2D: **Canvas**, **PixiJS**, **PhaserJS**
 - Gráficos 3D: **ThreeJS**, **BabylonJS**
- Eche un vistazo a **Qwik** y **Partytown**.
- **Houdini** - Aquí hay un buen **artículo introductorio a houdini**.
- **WebRTC** - WebRTC es un proyecto abierto y gratuito que proporciona a los navegadores y aplicaciones móviles capacidades de comunicaciones en tiempo real (RTC) a través de APIs sencillas. Los componentes WebRTC han sido optimizados para servir mejor a este propósito.
- **IDLs** - No todos los desarrolladores tendrán la necesidad de entender **IDLs**, pero hay un buen argumento para tener más desarrolladores de JavaScript conscientes de su existencia y, con suerte, la capacidad de hacer uso de ellos también.
- **WebAssembly**: WebAssembly es una tecnología relativamente nueva que permite ejecutar en la web código escrito en lenguajes distintos de JavaScript. Es un lenguaje ensamblador de bajo nivel con un formato binario compacto que se ejecuta con un rendimiento casi nativo y proporciona a lenguajes como C/C++ y Rust un objetivo de compilación para que puedan ejecutarse en la web.

Este roadmap.sh ha sido
traducido por rortegag.com

Los enlaces del documento
son de roadmap.sh y no están
traducidos