

Contenido

Guía para ser desarrollador web	3
Nivel Patito.....	4
Frontend	4
La trinidad	5
HTML	5
CSS	6
JavaScript.....	7
Bibliotecas y frameworks	8
Línea de comandos	8
Controlador de versiones.....	8
React.js	9
Next.js	9
Redux.js.....	10
SASS.....	10
Bootstrap	10
API.....	11
Animaciones 2D, 3D y eventos especiales.	11
Gsap	11
Three.js.....	11
SEO.....	12
Optimización del rendimiento web	13
Extra.....	13
Desarrollo y Producción.....	15
Dominio	15
Hosting	15
Backend.....	16
Bases de datos relacionales	16

Bases de datos no relacionales.....	17
Api de backend	18
Bibliotecas y frameworks	18
Express.js	18
Nest.js.....	19
Django.....	19
Spring boot	19
Asp.net	19
Go	19
Plus	19
TypeScript	19
Docker.....	20
Web3.js	20
Figma	20
Canva	20
Photoshop, Illustrator y Cualquier editor de video	20
¿Quieres crear un e-commerce?.....	20
CMS	20
Wordpress	21
Squarespace.....	21
Wix	21
Blogger.....	21
MailChimp, MailJet y Wufoo.....	21
END	22

Guía para ser desarrollador web

Bienvenidos sean todos a esta guía de desarrollo web, aquí podrán ver una ruta muy clara para poder convertirse en un desarrollador web desde 0, iré mostrándoles la ruta de aprendizaje que deben seguir y resumiré lo importante de cada etapa, también tratare de no hablar de forma muy técnica para que los nuevos de verdad se sientan más cómodos y no se confundan, esto con el deseo de que no se estanquen en el aprendizaje y puedan rápidamente desarrollar los conocimientos necesario para ser desarrollador web y ser capaces de desarrollar cualquier tipo de página web a nivel profesional, esta guía la estaré actualizando cada que vea necesario, corrigiendo o agregando nuevo contenido.

En varias ocasiones realizare menciones a los elementos importantes que deben estudiar, poniéndolos en **negrita**, estos elementos en negrita deberían buscarlos en la documentación de esa tecnología o directamente buscarlos en Google, si quieren encontrar contenido para aprender estas tecnologías de manera gratis, en YouTube encontraras videos completos que te enseñaran a usarlas, pero también está la opción de pagar por cursos en plataformas en línea, las que les recomiendo serian **Udemy**, **Platzi** y **Domestika**.

Con esta guía encontraran todo lo necesario para ser desarrolladores web al completo, el tiempo o velocidad en que puedan completar este proceso dependerá de cada persona y el tiempo que le dedique, siguiendo esta guía aceleraran mucho más rápido el proceso, sabiendo que deben estudiar realmente y no perdiendo el tiempo en cosas que muy posiblemente no les valla a servir.

Aprenderán muy rápido, pero para dominar al completo las tecnologías necesarias para ser desarrollador web deberán practicar mucho, recomiendo que creen muchas páginas web, que vayan implementando lo que aprenden y también que traten de copiar el diseño o funcionamiento de páginas web top del mercado, del nicho al que quieran dirigirse o simplemente páginas que les guste, de esta manera también irán aprendiendo cosas nuevas que seguramente no encontraran en esta guía, ya que hay cosas que solo se pueden aprender con experiencia y resolviendo problemas reales.

Nivel Patito

Empezaremos la guía sabiendo que hay ciertos requisitos mínimos que debemos tener, esta guía es para aprender desde 0 y convertirnos en desarrollador web, si ya tienes un conocimiento bueno manejando tu ordenador creando y manipulando archivos, sáltate este nivel, nos aseguraremos de que dominemos ciertos conceptos:

1. Debemos saber sobre **formatos de archivos**, es importante poder identificar y ver la diferencia entre un formato de imagen como .jpg, un formato de video como .mp4 o un formato de audio como .mp3, iremos viendo diferentes tipos de formatos a lo largo del aprendizaje, ojo con este punto ya que, en ocasiones, por no saber qué es esto, podría causarte problemas en el código y tu proyecto podría no funcionar
2. Debemos saber **crear y manipular carpetas en nuestro sistema operativo**, al hacer páginas web estaremos constantemente creando, eliminando y editando carpetas, estas contendrán nuestros archivos multimedia y todo el código que hace funcionar nuestras páginas web.
3. Debemos saber buscar nuestras dudas a través de internet, **Google** es el amigo número 1 de cualquier desarrollador, incluso desarrolladores profesionales están constantemente usando Google para resolver sus problemas en sus proyectos del trabajo o aprender nuevas tecnologías para aplicar en sus proyectos, también puedes hacer uso de las nuevas tecnologías como **ChatGPT**, que te ayudaran de manera muy clara fluida a encontrar la información que buscas y resolver problemas.
4. Necesitamos un nivel de inglés básico, eso para poder leer las documentaciones oficiales de las tecnologías que usaremos, estas estarán normalmente en inglés.

Frontend

Primeramente, aprenderemos las tecnologías que abarcaríamos como desarrolladores **frontend**, el frontend en resumidas cuentas vendría siendo esa parte de la página web con la que interactúan los usuarios, lo que el usuario ve en la pantalla, veamos cuales son las tecnologías que deberías empezar a dominar para realizar el frontend de tu página web, sigue esta lista en orden.

Te recomiendo que mientras estes aprendiendo uses un editor de código, te recomiendo **Visual Studio Code**, con esta herramienta es con la que estarás estudiando y construyendo el código de tu página web, como ultima recomendación antes de que empieces, trata de mientras vas aprendiendo y creando tus futuros proyectos, crea documentaciones, aprende a **documentar tus proyectos**, ahora que estas empezando, Tómalo con calma, pero al final cuando seas un desarrollador hecho y derecho tendrás que dominar eso.

La trinidad

HTML

Lo primero que siempre tendremos al momento de crear una página web es el **HTML**, es básicamente el esqueleto de nuestra página web, es lo que nos permitirá crear esos menús, secciones y botones que nuestra aplicación necesita, es muy sencillo y fácil de aprender, el formato de nuestros archivos **HTML** siempre será .html, te mencionare las etiquetas más importantes y que más se repiten al usar html:

1. `<html>`
2. `<head>`
3. `<meta>`
4. `<title>`
5. `<body>`
6. `<nav>`
7. `<ul>`
8. `<li>`
9. `<footer>`
10. `<section>`
11. `<header>`
12. `<article>`
13. `<div>`
14. `<form>`
15. `<button>`
16. `<iframe>`
17. `<h1>`
18. `<p>`
19. `<span>`
20. `<img>`
21. `<svg>`
22. `<a>`

23. <main>

24. <input>

25. <table>

26. <audio>

27. <video>

28. <style>

29. <script>

Puedes buscar y aprender a usar todas estas etiquetas en la siguiente documentación:

<https://www.w3schools.com/html/default.asp>

CSS

Si el **HTML** es el esqueleto de nuestra página, el **CSS** es lo que vendría siendo la piel y nuestra ropa, le darás forma, color y posición a los elementos en pantalla, el formato de un archivo **CSS** es .css lo principal que debería aprender de **CSS** es lo siguiente:

- 1. Selectores**
- 2. Width**
- 3. Height**
- 4. Color**
- 5. Position**
- 6. Display**
- 7. Transition**
- 8. Transform**
- 9. Font-size**
- 10. Font-family**
- 11. Font-weight**
- 12. Line-height**
- 13. Overflow**
- 14. Margin**
- 15. Padding**

- 16. **Min-widht**
- 17. **Max-widht**
- 18. **Min-height**
- 19. **Max-height**
- 20. **Uso de unidades relativas**
- 21. **Media query**

El último punto es importantísimo, ya que debes aprender a cómo usar las media query para que tu página web sea responsive, ósea, esta se pueda adaptar a todos los dispositivos posibles según tu proyecto, de lo contrario tu página web puede romperse y verse fatal en celular o pc, puedes buscar y aprender a usar todos estos conceptos en la siguiente documentación:

<https://www.w3schools.com/css/default.asp>

JavaScript

Aquí tenemos al que será nuestro lenguaje de programación estrella, **javascript** será aquello que nos permitirá agregarle esa lógica que necesita nuestra página web, aunque es verdad que puedes no usar javascript para crear páginas web, también es verdad que si no lo implementas, te veras limitado en muchas ocasiones por no poder agregarle las funcionalidades extra que necesita tu página web, javascript es algo que siempre vas a tocar directa o indirectamente y es de los lenguajes de programación más solicitados en el mercado laboral, con esta poderosa herramienta no tendremos límites al momento de trabajar en nuestros proyectos, el formato de nuestros archivos **javascript** serán .js, sirve para manipular datos, manipular la información en pantalla, controlar múltiples eventos en nuestra página web, realizar transiciones o animaciones complejas, crear video juegos para la web, realizar conexiones a **APIs** o **base de datos**.

A continuación, veremos lo principal que debes aprender para usar javascript en tus proyectos web:

- 1. **Variables**
- 2. **Operaciones**
- 3. **Condicionales**
- 4. **Ciclos**
- 5. **Funciones**
- 6. **Clases**
- 7. **Objetos**

8. Herencia

9. Asincronismo

10. Manipular el contenido, propiedades y clases de una etiqueta html con javascript

Puedes buscar y aprender a usar todos estos conceptos en la siguiente documentación:

<https://www.w3schools.com/js/default.asp>

Bibliotecas y frameworks

En esta sección vamos a ver herramientas que nos van a facilitar la vida a la hora de crear nuestras aplicaciones web.

Línea de comandos

Debemos tener un conocimiento básico de **líneas de comando**, como desarrolladores web estaremos interactuando con la línea de comandos siempre, ya sea para Windows, Linux o Mac, recomiendo buscar una pequeña guía básica dependiendo del sistema operativo que estes usando, cosas principales que debes dominar son:

1. **Movilizarte por las carpetas de tu ordenador haciendo uso de la terminal**
2. **Creando, editando o eliminando archivos desde la terminal**
3. **Limpiar la vista en la terminal**

De hecho, con esos 3 puntos ya serian suficiente por si quieres avanzar mas rápido, pero eso sí, saber bien a usar la línea de comando te dará un plus profesional muy notable.

Controlador de versiones

Cuando estamos trabajando en nuestros proyectos, es importante tener un control de versiones, poder ir actualizando y respaldando cada paso que estamos realizando en nuestro proyecto, los controladores de versiones nos ayudan a esto también a trabajar más fácil con colaboradores en nuestro proyecto, teniendo el orden y control de lo que está pasando con nuestro código todo momento, en este caso la recomendación será aprender a usar **github**, aquí es donde ahora en adelante estarás subiendo tus proyectos, respaldándolos y teniendo control total de cada etapa del desarrollo de tu aplicación, para entenderlo mejor es entrar en la práctica, lo principal que deberás aprender para iniciar en git será:

1. Instalar **git bash** <https://git-scm.com/downloads>
2. Crear una cuenta en **github** <https://github.com/>
3. Aprender que es un **repositorio**

4. Aprender que son las **ramas**
5. Aprender los comandos básicos para saber cómo configurar y progresar con tu proyecto usando git <https://gist.github.com/dasdo/9ff71c5c0efa037441b6>

React.js

Ahora sí, el momento de crear nuestra aplicación web profesional ha llegado, y para eso les recomendaré usar **React.js** con ella vamos a crear toda nuestra interfaz, creando páginas web de forma sencilla haciendo uso de componentes y al final poder mandar nuestra página web a producción en su versión más optimizada haciendo uso de unos cuantos comandos.

Existen otras librerías o frameworks que sirven para los mismos propósitos que **React.js**, are mención se **Angular.js** y **Vue.js**, estas tres herramientas tienen sus ventajas y desventajas, los invito a investigar cada una y decidir por ustedes mismos cuál de estas les gusta más, para esta guía nos enfocaremos en **React.js**.

Para empezar a jugar con **React.js**:

1. Vamos a necesitar instalar **Node.js** en su versión **LTS** <https://nodejs.org/es/>
2. En la línea de comandos pondrás el siguiente comando **npx create-react-app my-app** esto creará nuestra página web con todo lo necesario
3. Luego pondrás **cd my-app**
4. Ahora iniciaremos nuestra página web con **npm start**, esto nos desplegará nuestra página en el navegador por defecto que tengamos
5. Aprende que son los componentes en react
6. Aprende que son los **hooks** en react
7. Aprende a usar **useRef**, **useEffect**, **useState**, **useReducer** y **useContext**
8. Aprende a hacer una conexión a un **API** y usar esa información en tu página web

Puedes buscar y aprender a usar todos estos conceptos en la siguiente documentación:

<https://es.reactjs.org/docs/create-a-new-react-app.html>

Next.js

Esta herramienta vendría a extender lo que es react, con ella podremos hacer mucho más fácil nuestro desarrollo web, aquí podrás seguir usando **react.js** como tal, pero la sintaxis y la organización de carpetas cambiará un poquito, **next.js** también es muy popular por su sencillez y dinámico enrutamiento, para crear blogs, APIs de manera muy sencilla y

optimizar tanto el **SEO** como el peso de nuestras imágenes haciendo uso de su etiqueta especial **<Image>**.

Puedes ver esta guía en la siguiente documentación oficial:

<https://nextjs.org/docs/getting-started>

Redux.js

Esta herramienta es muy utilizada en proyectos como react.js y angular.js, nos sirve esencialmente para poder controlar el estado en nuestras aplicaciones, es posible que necesites interactuar con esta herramienta dependiendo del proyecto que estes manipulando.

Te recomiendo investigar la siguiente documentación de **Redux** para conocer sobre su implementación he impacto:

<https://redux.js.org/introduction/getting-started>

SASS

Esto no es obligatorio, pero probablemente te solucione la vida cuando veas que tu hoja de **CSS** tiene más de 5000 líneas y pierdes mucho tiempo buscando lo que necesitas editar para darle mantenimiento a los estilos de tu página, por lo que un preprocesador de **CSS** como **SASS** será tu mejor aliado, ya que con **SASS** podrás fragmentar tu **CSS** en distintas hojas y **SASS** las unirá en un único **CSS** por ti automáticamente, también **SASS** agrega propiedades y características de programación para que las uses en **CSS** facilitando algunos procesos a la hora de crear estilos, es fácil de aprender por que básicamente **SASS** sigue siendo **CSS**.

Te invito a leer la documentación de **SASS** a continuación:

<https://sass-lang.com/documentation/>

Bootstrap

Si quieres simplificarte un poco más la vida al momento de montar componentes, no trabajar mucho en el **responsive desing** o buscas una alternativa diferente a la hora de dar estilos a tu página web, pues **bootstrap** es tu herramienta, es muy fácil de usar y pueden empezar y probar rápido simplemente copiando y pegando directamente el código de la página oficial a tu proyecto, eso sí, claro que es personalizable, existen otras páginas como bootstrap que cumplen casi con las mismas funciones, té mencionare tres que te pueden interesar **Materialize**, **Tailwindcss** y **Chakra UI**.

Empieza probando bootstrap mirando su documentación:

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

API

Una cosa es segura, siempre en algún momento vas a tener que necesitar hacer uso de una **API**, de estas podremos hacer solicitudes y extraer información para ser usada y moldeada en nuestras páginas web, esto nos puede ser muy útil, existen **APIS** para todo propósito, para el área de economía, para saber el clima, para listas de productos, para hacer testeos en nuestras aplicaciones, para criptomonedas, la forma de interactuar con estas **APIS** dependerá de cada proyecto y la **API**, podemos aprender y practicar como usarlas con **APIS** populares usadas para este propósito, aprende a hacer una conexión con estas **APIS**, existen de paga y otras gratis, les recomiendo aprender usando estas:

1. **The Rick and Morty API** <https://rickandmortyapi.com/>
2. **PokeApi** <https://pokeapi.co/>
3. **Nasa** <https://api.nasa.gov/>

Animaciones 2D, 3D y eventos especiales.

Algo además de un buen diseño y optimización de nuestras páginas que no podemos dejar de lado, es realizar Animaciones de calidad y manipularlos según la ocasión, esto llevara nuestro diseño web a otro nivel, las herramientas que mencionare a continuación son clave, aunque no son las únicas, son de las más usadas a la hora de crear páginas webs alucinantemente increíbles.

Gsap

Sin duda alguna una herramienta que no debería faltar en tu stack, con ella podremos hacer todo tipo de animaciones en nuestra página web ya sea para manipular elementos 2d o 3d, crear funciones que indiquen como cuando ejecutar estas animaciones de forma muy fácil y también con ayuda de su plugin **ScrollTrigger**, podemos fácilmente manipular animaciones que se activan con el movimiento del scroll que este haciendo los usuarios en la página web.

Aprende a usar esta herramienta y como implementarla a cualquier proyecto en sus respectivas documentaciones:

Gsap: <https://greensock.com/gsap/>

ScrollTrigger: <https://greensock.com/scrolltrigger/>

Three.js

Posiblemente si has visto alguna página con elementos 3d, animaciones increíbles o portafolios a otro nivel, es porque seguramente están usando **three.js**, esta herramienta nos permite manipular objetos 3d y visualizarlos en nuestras páginas web, puedes importar tus modelos 3d desde **blender** y demás aplicaciones para el modelado 3d, se puede combinar

con **Gsap** para hacer cosas realmente alucinantes, para usar esta herramienta solo basta leer la documentación, pero eso sí, si tienes conocimientos básicos de modelado 3d, te será más fácil entender y usar esta herramienta, puedes intentar aprender lo básico usando **blender**, ya que es una herramienta de uso gratuito.

Aprende esta hermosa herramienta haciendo uso de su documentación oficial:

<https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>

SEO

Optimización en motores de búsqueda, eso es a lo que nos referimos con **SEO**, si quieres que tu página web se posicione por encima de otras páginas web o contenidos en los buscadores para alcanzar más fácil ese público que buscas, debes dominar por completo el **SEO**, existen muchas formas de evaluar y aplicar el **SEO** en tu página web, además de que debes usar varias herramientas y cumplir varios pasos, un buen **SEO** empieza desde el código.

1. Primero aprende a evaluar el estado del **SEO** actual de tu página, para esto vas a ir a aprender a usar la siguiente herramienta **Lighthouse de Google** eso realizara un análisis automático del **SEO** de tu página web si se lo indicas
2. Revisar que tus paginas tengan incluido la meta etiqueta de descripción `<meta name="description" content="text">` y la etiqueta de título `<title>Document</title>` ya que estas representan la descripción y título de la página que encontrara el usuario en los buscadores, también tener en cuenta que solo se recomienda tener una sola etiqueta `<h1>` por página y debe ser un título importante ya que será tomado en cuenta en el **SEO**
3. Crear un archivo **site.xml**, esto ayudara al navegador a saber cuáles son nuestras URL indexables, de paso también investiga sobre el uso del archivo robots.txt
4. Usa Google Search para optimizar y analizar el **SEO** de tu página web
5. Según las herramientas que hallas usado para crear tu página web, deberás investigar cómo aplicar o mejorar el SEO con estas, por ejemplo, Como mejorar el **SEO** con **React.js** usando **Helmet**
6. Utiliza **Google Analytics** en tu página web, lleva un estudio de tu tráfico y estudia nuevas oportunidades de mejoras en tu web
7. El **SEO** toma su tiempo, por lo que no esperes ser la página top 1 en primer día, toma tiempo y análisis, por lo que también recomiendo estudiar a tu competencia dependiendo de tu proyecto, así entender que hacer para lograr subir tu posicionamiento en la web

Optimización del rendimiento web

Tener una página web optimizada será clave para que tu web, no solo que no tenga problemas de accesibilidad, sino porque, si tu página no está optimizada, carga lento o en ocasiones se rompe, esto puede afectar negativamente el tráfico de tu página web y además el **SEO**.

Te daré ciertas recomendaciones para mejorar la optimización de tu página web, pero igual te invito a investigar, lo que puedes hacer como resumen seria lo siguiente:

1. Aprende a usar **Lighthouse de Google** eso realizara un análisis automático del estado del rendimiento de tu web, tanto para móviles como ordenadores si se lo indicas
2. Revisa que tus archivos multimedia sean lo más ligeros posibles, por ejemplo, que tus imágenes no pesen mucho, esto puede causar que la página web cargue lento
3. Verifica si hay código en tu aplicación que no estas usando nunca, esto puede causar que la página web cargue lento, esto normalmente suele pasar sobre todo cuando no tenemos **buenas prácticas de programación**
4. Mira tú **CSS**, identifica si tienes selectores muy complejos y trata de mejorarlo, entre más complejo son los selectores en tu hoja de **CSS**, más lento cargara tu página esos elementos de manera correcta
5. Aprende a usar herramientas de testing, como lo son **jest.js** o **selenium**

Estos son algunos consejos que seguro te ayudaran, investiga que otras herramientas o técnicas te pueden ayudar a optimizar tu página web.

Extra

Te comparto algunos enlaces a herramientas que te facilitaran el desarrollo y muy fáciles de usar:

1. <http://stickerjs.cmiscm.com/>
2. <https://ireade.github.io/inlinetweetjs/>
3. <https://materializecss.com/>
4. <https://vincentgarreau.com/particles.js/>
5. <https://animejs.com/>
6. <https://mojs.github.io/>
7. <https://maxwellito.github.io/vivus/>
8. <https://bennettfeely.com/clippy/>

9. <https://smooth.ie/blogs/news/svg-wavey-transitions-between-sections>
10. <https://uigradients.com/#Dull>
11. <https://www.cssmatic.com/box-shadow>
12. <https://animate.style/>
13. <http://imagehover.io/>
14. <https://loading.io/css/>
15. <https://cssgradient.io/>
16. <https://jonsuh.com/hamburgers/>
17. <https://kenwheeler.github.io/slick/>
18. <https://aframe.io/examples/showcase/a-painter/>
19. <https://github.com/jgthms/lavalamp.js>
20. <https://usablica.github.io/progress.js/>
21. <https://github.com/VPenkov/okayNav>
22. <https://kazzkiq.github.io/balloon.css/>
23. <https://tailwindcss.com/>
24. <https://chakra-ui.com/>
25. <https://www.chartjs.org/>
26. <https://sweetalert.js.org/>
27. <https://sweetalert2.github.io/>
28. <https://sortablejs.github.io/Sortable/>

Desarrollo y Producción

Digamos que ya hemos terminado de crear nuestra aplicación web, lo que concluiría con nuestra **etapa de desarrollo** y ahora queremos que la pagina este publica para nuestros usuarios, así pasando a **la etapa de producción**, lo que debemos hacer primero es conocer los siguientes conceptos.

Dominio

El **dominio** es aquel nombre único con el que nuestra página se va a identificar al público, un ejemplo es **Google.com**, siempre que queramos mandar a producción nuestra página web lo indispensable es adquirir un dominio para nuestra página, si no tienes claro que es dominio, te invito a investigar **que es un dominio y las partes que componen un dominio**.

Para adquirir un dominio, lo más normal es comprarlo por medio de algún proveedor de dominios, te en listare algunos de los servicios usados para comprar un dominio:

1. <https://www.godaddy.com/es>
2. <https://www.dondominio.com/>
3. <https://www.bluehost.com/domains>

Para poder conectar tu dominio a tu página web, deberás primero tener subida tu aplicación en un **Hosting** he investigar como configurarlo o redireccionar correctamente tu dominio a tu página web, es normalmente casi lo mismo para cada caso, pero dependiendo del hosting o proveedor de dominio, el proceso podría ligeramente cambiar, una vez configurado tu dominio a tu página web este puede durar alrededor de 72 horas más o menos en verse aplicado los cambios.

Hosting

Siguieres subir tu página web y ligarla a un **dominio**, tendrás que pagar por un servicio de **hosting**, este es el servidor donde estará alojada tu página web, hay mucho servicios populares donde puedes subir y configurar lo necesario para tu página web, te daré unas recomendaciones y te comento que deberías revisar cual de estos te conviene, algunos de estos servicios de **hosting** vienen con dominio gratis por un año y viene configurado para que de una estes al aire con tu dominio, también dan **SSL GRATIS**, investiga que es **SSL**, te ayudara en cuanto a seguridad con tus usuarios y en el **SEO**:

1. <https://www.hostinger.es/>
2. <https://www.hostgator.com/>
3. <https://aws.amazon.com/es/free/webapps/>

Backend

Ahora veremos la otra cara de la moneda, en este caso, las tecnologías para ser un desarrollador **backend**, a diferencia del **frontend**, con el **backend** nos referimos a la parte de nuestra aplicación que vive del lado del servidor y que el usuario por ende no ve, básicamente aquí veremos todo lo que tiene que ver con la base de datos de nuestra página web, de donde nuestra página web tomara la información que necesite manipular para mostrar o no al usuario en nuestro **frontend**.

Como desarrollador web podrás elegir entre estos dos caminos, puedes aprender solo **frontend** o solo **backend**, estos dos suelen trabajar juntos dependiendo del proyecto, pero también puedes ser ambos y convertirte en un desarrollador web **Full Stack**, eso sí, en ese caso, será más demandante el tiempo de estudio que deberán dedicar.

Dos cosas importantes que mencionar, estamos claro que nuestra **base de datos** es donde almacenamos la información, para luego manipularla y usarla en conjunto con nuestras aplicaciones, en nuestro caso con la página web que tengamos, también mencionar que para manejar estas bases de datos es muy recomendable aprender sobre **línea de comandos**.

Bases de datos relacionales

En este tipo de bases de datos se sigue el modelo relacional, los datos se guardan en tablas y cada tabla tiene un ID único, si quieres aprender a usar una base de datos relacional, deberás aprender **SQL** que es un lenguaje de consulta estructurado, con el podremos manipular nuestra base de datos para cumplir con nuestros objetivos, primero debemos elegir una base de datos, te recomendaré las siguientes **MySQL, Oracle, SQL Server**, existen más por si quieres investigar y encontrar una que se adecue a tu aplicación, ahora, sigue esta pequeña ruta para que inicies en el mundo de las bases de datos relacionales:

1. Si ya elegiste tu base de datos, instálalo para practicar el **SQL**
2. Aprende como crear y administrar **usuarios**
3. Aprende como crear y administrar **roles**
4. Aprende como usar **Create** *con esto crearas tu primera tabla*
5. Aprender cómo usar **Select**
6. Aprender cómo usar **Insert**
7. Aprende como usar **Update**
8. Aprende como usar **Delete**
9. Aprende como usar **AS**
10. Aprender a usar **WHERE, ORDER y GROUP**

11. Aprende como hacer **Join**
12. Aprende como crear una **llave única** o **ID**
13. Aprende a crear **llaves foráneas**
14. Aprende a crear y aplicar **Secuencias**
15. Aprende como usar **Alter**
16. Aprende a crear **Vistas**
17. Aprende sobre los **atributos de los campos**
18. Aprende sobre la **normalización en las bases de datos y sus formas**
19. Aprende a crear **Triggers**
20. Aprende a crear una table **pivot**
21. Aprende todo sobre **PL/SQL**
22. Aprende sobre **Querys Analíticos**
23. Aprende a crear **respaldos de seguridad** para tu base de datos
24. Aprende a **exportar he importar** tus bases de datos
25. Aprende **como hacer una conexión de la base de datos** que elegiste a tu proyecto

Esto es lo básico que necesitas para crear tu primera base de datos, eso sí , hay mucho más, ya depende de vos hasta donde llegas con el tema, según la base de datos que elijas puede que cambien un poco la sintaxis, casi nada, al final de todo es **SQL** si aprendes a usar una base de datos, dominar otra será pan comido.

Bases de datos no relacionales

Estas bases de datos vendrían siendo las bases de datos **NoSQL**, aquí no usaremos **SQL** como tal y la información que guardemos no necesariamente debe seguir el modelo relacional, este tipo de base de datos son muy flexibles y a diferencia de las bases de datos **SQL**, aquí en vez de tablas, manejamos Colecciones y documentos, las bases de datos **SQL** Y **NoSQL** tienen sus ventajas y desventajas, para elegir una adecuada para tu proyecto, te invito a investigar sobre el tema, te recomendaré dos base de datos no relacionales **Mongo db** y **Cassandra**.

Por lo general la base de datos **NoSQL** suelen ser más fácil de aprender que una base de datos **SQL**, con estos pasos podrás entrar tranquilamente a esta base de datos y probar en tus proyectos:

1. No es obligatorio, pero seguramente si estudias sobre que es un archivo **JASON** para almacenar datos, como manipularlo y usarlo, puede que se te facilite más al momento de empezar en el **NoSQL**
2. Aprende sobre **usuarios y roles**
3. Aprende sobre la creación y manipulación de una **Colección**
4. Aprende sobre la creación y manipulación de un **Documento**
5. Aprende a crear **respaldos de seguridad** para tu base de datos
6. Aprende a **exportar he importar** tus bases de datos
7. Aprende como **conectar la base de datos** que elegiste a tu aplicación

Esto es lo básico que necesitaras, también hay muchas más bases de datos no relacionales, por si gustas investigar, puede que otra te guste más o se acomode mejor a tu caso.

Api de backend

Existen **APIs** que te pueden ayudar a la hora de trabajar en tu página web, estas nos pueden ayudar a interactuar con los servidores de nuestra base de datos entre otras cosas, o darte información de prueba para testear tu página web, te comparto una pequeña lista de **APIs** que puedes probar en tus proyectos:

1. <https://www.programmableweb.com/api/stackapi-rest-api>
2. <https://firebase.google.com/docs/reference/rest/database>
3. <https://tradematic.cloud/>

Bibliotecas y frameworks

Bueno aquí veremos un poco de todo, las siguientes herramientas son una recomendación para explorar, no es necesario que las aprendas todas, ya que son algo que requerirás dependiendo del proyecto.

Express.js

Es un framework que nos ayudara a realizar el backend de nuestra aplicación que va a conectarse a nuestra página web, debes tener **node.js** instalado para usarlo.

Nest.js

Es un framework que nos ayudara a realizar el backend de nuestra aplicación al igual que express.js, te ayuda a construir tu backend de forma muy fácil y usaras **typescript** para trabajar con él, debes tener **node.js** instalado para usarlo.

Django

Es un poderosísimo framework de **Python** que nos permitirá construir de forma rápida tanto el frontend como el backend, permitiendo que los dos vivan juntos, igual puede usarlo si quieres solo para el backend y usar otra herramienta para el frontend con esta herramienta tendrás que programar en el lenguaje de programacion **Python**.

Spring boot

Nos permitirá crear aplicaciones autocontenidas, esta herramienta es muy poderosa y te ahorra mucho tiempo de desarrollo al usarla, nos permite compilar una página web como un archivo.jar que podremos ejecutar como si fuera una aplicación de **java**!, si, puedes también trabajar tu frontend utilizando el lenguaje de programación **java**.

Asp.net

Es un entorno de desarrollo web también muy usado, puedes crear tus páginas web usando el lenguaje de programación **C#**, tiene múltiples herramientas que facilitaran tu conexión a base de datos sin mucho esfuerzo de más.

Go

Es un lenguaje de programacion que nos permite hacer muchas cosas, trabajar del lado del servidor, crear sitios web, web scraping, **APIS** y más, puedes darle una oportunidad a esta herramienta, puede que te sorprendas y sea lo que estas buscando.

Plus

Hay ciertos conocimiento plus que te ayudaran a ser un mejor desarrollador, te mencionare una lista de tecnologías que te recomiendo para llevar tanto en el frontend y el backend a otro nivel.

TypeScript

Este lenguaje de programación es muy popular y es fácil de aprender, porque es básicamente **javascript**, pero con algunas mejoras, te permite reducir los errores o bugs de tu aplicación, mejoraras en buenas prácticas y le agrega a javascript el tipado de datos, si lo pruebas posiblemente te vas a enamorar.

Docker

Nos permite desplegar aplicaciones mucho más rápido, mejora y reduce los problemas al desarrollar con equipos de trabajo, una vez creas los populares contenedores de **Docker**, tu código servirá siempre igual independientemente en qué lugar lo estes corriendo, posiblemente sea una herramienta que vean muy seguido en el mundo laboral.

Web3.js

Nos ayuda a crear aplicaciones en la web 3.0, excelente para proyectos basados en el mundo de las crypto monedas.

Figma

Es un editor grafico que te permitirá maquetar tus aplicaciones de forma fácil y rápida, excelente para crear el diseño, wireframes y prototipo visual para tu frontend.

Canva

Es una página donde desde la web puedes crear diseños muy profesionales fácilmente para múltiples propósitos, perfecto para diseño.

Photoshop, Illustrator y Cualquier editor de video

Tener conocimientos de estas herramientas de diseño será excelente, sobre todo si eres una persona independiente, podrás hacer poderosos diseños de interfaces sin recurrir a un diseñador.

¿Quieres crear un e-commerce?

Aprende a usar frameworks o librerías que puedan ayudarte a crear la base para vender tus productos o servicios, por ejemplo: **paypal SDK**, **Stripe** o usando algún **CMS**.

CMS

El sistema de gestor de contenidos nos permite crear páginas web de todo tipo en cuestión de unos cuantos clics, incluso sin tener mucha idea de programación, esto te ahorrará tal vez no horas, no días, si no hasta meses de desarrollo con ayuda de sus herramientas que nos permiten hacer las cosas más fáciles a la hora de crear nuestro frontend, además de no tener que pensar casi que ni siquiera en el backend, eso sí, si no tienes conocimientos en **HTML**, **CSS** y **JavaScript**, tu página web va a verse muy limitada a lo que solo ofrecen estas herramientas, te recomendaré varias herramientas para que las pruebes de primera mano, el tiempo de aprendizaje de estas es muy rápido, puede incluso de un día o solo unos cuantos, dependiendo de que necesites hacer.

Wordpress

Es un **CMS** de código abierto, esto le da un poco de ventaja con respecto a los otros CMS, ya que tendremos más control de nuestro proyecto y lo podremos personalizar muchísimo más, para crear tu frontend puedes usar el plugin **Elementor**, este sería un editor que te permite con unos clic darle la forma y diseño a tu página web, te permite también usar bloques de código para que pongas tu código personalizado en tu web, también si quieres crear un **e-commerce**, puedes instalar el plugin de **WooCommerce** para preparar fácilmente tu tienda, **wordpress** tiene muchos **plugins** con los que puedes gestionar los certificados **SSL**, el **SEO**, la **optimización y la seguridad** con unos cuantos clic, si deseas usar **wordpress**, puedes hacerlo directamente en **wordpress.com** y ver sus tarifas, pero si quieres puedes descargar **wordpress** desde la página oficial **wordpress.org**, de esta forma incluso te saldrá más económica realizar y llevar a producción tu página, por ejemplo puedes usar el servicio de **Hostinger** que te provee de servers optimizados para tu página **wordpress**.

Squarespace

A diferencia de **wordpress**, es de código cerrado, por lo que no tendrás tantas libertades, pero igual podas crear todo tipo de páginas, blogs y e-commerce, aunque es de código cerrado, a través de su editor puedes agregar código personalizado a través de su bloque de código y otras opciones, **Squarespace** suele tener tarifas un poco más elevadas por lo que debes revisar en su página oficial **squarespace.com** y probar.

Wix

Es la más fácil de manipular y aprender, puedes igual hacer blog, e-commerce y más, su editor te permite también usar bloques de código para más personalización, además a diferencia de **Squarespace** y **Wordpress**, puedes hacer muchas más cosas sin saber nada de **HTML**, **CSS** y **JavaScript**, ya que su editor es muy poderoso y te permite configurar y editar aspectos visuales con demasiada libertad.

Blogger

Este es otro recomendado, más enfocado a crear blogs, como lo indica su nombre.

MailChimp, MailJet y Wufoo

Son servicios que te permitirán generar formularios para tus páginas web, este hecho por **CMS** o no, y puedes editar su diseño con código, solamente es crear el formulario, generar el código que debes insertar en tu página y este formulario quedara listo y funcional para cumplir con su propósito, muy útiles para el área de marketing por su conjunto de herramientas.

END

Existen otras herramientas y tecnologías para potenciar tu lado backend o frontend, pero para alivianarte el camino te comparto que en realidad no es necesario que sepas todas, todo depende del proyecto en el que estes trabajando o lo que necesite la empresa que te esté contratando, por ende, paso de mencionarlas por el momento, así que, si en un momento te toca aprender una nueva tecnología, recuerda, Google es tu amigo.

Con esto concluimos con la guía para ser un desarrollador web, espero esto te haya servido de ayuda, ya sea para aclarar el panorama o aprender del mundo del desarrollo web, **esta página web estará siendo actualizada** cada cierto tiempo, te invito a ver las guías **Quick start** para ciertas tecnologías que estaré subiendo aquí mismo, con el fin de iniciar, aprender fácil y rápido para usarlas en tus proyectos.

Autor: Brandon James Grimaldo Moscote