

# Aprender a programar desde cero: consejos y recursos

Hola gente, ¿cómo están? ¡Bienvenidos de vuelta a un nuevo artículo! En el día de hoy, quiero hablarles un poco sobre cómo comenzar a programar. Si tuviera que hacerlo de vuelta, mi objetivo acá no es criticar todas las cosas que hice hasta ahora o decir qué mal lo hice en estos aspectos, sino simplemente intentar dar un punto de vista objetivo de todas las cosas que yo he hecho en estos casi dos años programando. ¿Qué cosas he aprendido? ¿Qué cosas haría diferente? ¿Qué cosas agregaría para poder darles un feedback bastante crítico que les pueda servir y que ustedes lo pueden implementar en su día a día?

## El diseño web como punto de partida

Como mencioné en otro artículo, yo aprendí a programar o mis primeros pasos en la programación fueron con el diseño web, y esto creo que no es algo que yo haya hecho mal. De hecho, creo que es un paso que muchos de ustedes deberían tomar. Creo que el diseño web es algo que está bastante bueno como para empezar, es algo que puedes ver los resultados de manera visual bastante rápido y no tienes que entrar en cosas muy complejas desde el principio. Entonces, creo que el diseño web fue algo que hice bien, pero lo que sí me arrepiento un poco es de no haber aprendido antes lógica de programación.

## La importancia de la lógica de programación

Me quedé mucho tiempo en HTML y CSS, y si bien hice algo de JavaScript, no lo hice muy a fondo. Cuando tuve que tomar clases de programación en la universidad, me costó un montón

al principio. ¿Por qué? Porque en la universidad yo empecé con C++, que no es el lenguaje más amigable para una persona que empieza a programar. C++, al ser un lenguaje de nivel bajo, tiene una sintaxis más compleja y difícil de entender que un lenguaje como Python, por ejemplo.

Lo que pasa es que cuando uno tiene buena lógica de programación, por más que se encuentre con un lenguaje de nivel bajo, puede darse cuenta de hacia dónde va el código y cómo funciona, y así trabajar con él. En mi caso, no tenía una buena lógica de programación y, por lo tanto, no entendía el código, no sabía cómo hacer las tareas y me costaba mucho seguir la documentación. Por eso, les recomendaría que aprendan lógica de programación desde el principio.

Esto es muy importante no solo para poder crear programas de manera mucho más rápida, sino también para leer e interpretar código de manera más eficiente. Leer y aprobar código es una labor frecuente de un programador, y cuando nuestra lógica de programación está bien desarrollada y entendemos para qué sirve cada cosa dentro del programa y del lenguaje de programación en particular que estemos usando, nos resultará mucho más fácil realizar las tareas que tengamos que hacer y no perderemos tiempo en cosas innecesarias, como leer mucha documentación o pedir ayuda a muchas personas o fuentes para poder realizar un trabajo.

## **Git y GitHub: herramientas fundamentales**

Otra cosa de la que me arrepiento mucho es no aprender bien cómo se usa Git y GitHub. Esta es una herramienta fundamental para un desarrollador de software. En GitHub, vas a guardar casi todo tu código de los proyectos que vayas a hacer, y es muy importante porque puedes dejar una copia de tu código y puedes ir actualizándolo en base a cómo va evolucionando tu proyecto.

Además, en las empresas en las que uno puede trabajar, como en mi caso que estoy actualmente trabajando como ingeniero de software en pasantía acá en Estados Unidos, las empresas tienen repositorios en GitHub. Si no sabes cómo utilizar Git o cómo enviar y traer código desde esos repositorios hacia tu computadora para poder trabajar localmente, te resultará muy complicado trabajar en equipo, ya que casi todas las empresas están usando GitHub o alguna otra herramienta de control de versiones. También tendrás dificultades con tareas como aprobar «pull requests», que básicamente son revisiones de código, entre muchas otras cosas.

## **JavaScript: aprender las bases antes de saltar a un framework**

Otra cosa de la que me arrepiento mucho es no aprender bien JavaScript. Esto está relacionado a lo que es la lógica de programación que mencioné anteriormente, pero más enfocado al frontend. Me apuré mucho en saltar a un framework de frontend de JavaScript, como podría ser React o Vue, sin haber comprendido las bases de JavaScript. Esto me ocasionó muchos problemas al trabajar en proyectos, tanto personales como reales.

Me ha sucedido mucho no saber cómo seguir con el código, resolver un error o ni siquiera saber de dónde venía el error. Y esto era porque no tenía claras las bases de JavaScript. Tuve que perder mucho tiempo preguntando a otras personas, buscar en documentación o preguntar en foros cómo solucionar los errores que tenía, algo que se habría solucionado simplemente leyendo documentación de JavaScript, la cual no había leído antes.

Por eso, mi recomendación personal es que no salten a un framework de JavaScript sin comprender bien JavaScript y, sobre todo, sin haber desarrollado varios proyectos en JavaScript. Hagan proyectos, estudien las documentaciones y,

cuando se sientan seguros, pasen a un framework.

## **No quedarse solo con lo que se aprende en la universidad**

Otra recomendación personal que les puedo dar es que, si están estudiando ingeniería de software en la universidad o alguna carrera relacionada a la tecnología, nunca se queden solo con lo que ven en la universidad. A mí me ha pasado muchas veces que tenía mucha carga horaria y muchos trabajos en la universidad, y si bien me ha aportado conocimientos técnicos que no hubiera tenido de otra forma, creo que es muy importante hacer muchas cosas fuera de la universidad.

Si uno se queda solo con lo que aprende en la universidad, será muy difícil conseguir un trabajo en la industria del desarrollo, ya que las tecnologías que usa el mercado usualmente no son enseñadas o hacen énfasis en la parte práctica de programación. Por lo tanto, les recomiendo que hagan cursos, que hagan proyectos. Los cursos que tienen proyectos integrados están muy buenos, ya que la persona que lo hace te va explicando cómo se hace el proyecto y para qué sirve cada cosa. Es muy importante que siempre estén haciendo algo, no importa qué tan pequeño sea. Aprendan una nueva cosa cada día, pónganlo en un proyecto, desarrollen algo pequeño.

La idea no es hacer un proyecto gigante que revolucione el mercado, sino hacer proyectos continuamente, aprender cada día y mejorar. Si siguen este enfoque, con constancia, motivación y disciplina, podrán alcanzar sus metas y convertirse en desarrolladores en el área que les guste. No se asusten si una tecnología les parece compleja; crean en sí mismos y repitan todos los días, intentando ser un poco mejor que el día anterior.

Bueno, gente, eso era todo por hoy. Espero que les haya gustado el artículo y que les sirva para su camino en la

programación. No olviden suscribirse, compartir el artículo y dejarme sus comentarios. También pueden seguirme en Instagram para obtener más información y contenido relacionado. ¡Nos vemos en la próxima!

## Tabla Resumen

Puntos clave
Diseño web como punto de partida
Importancia de la lógica de programación
Herramientas fundamentales: Git y GitHub
Aprender las bases de JavaScript antes de saltar a un framework
No quedarse solo con lo que se aprende en la universidad

## Preguntas frecuentes

### 1. ¿Es necesario aprender diseño web antes de programar?

Si bien no es necesario, el diseño web puede ser un buen punto de partida para comenzar a programar, ya que permite ver resultados visuales rápidos y no se requieren conocimientos muy complejos desde el principio.

### 2. ¿Por qué es importante aprender lógica de programación?

La lógica de programación es fundamental para poder comprender y crear programas de manera eficiente. Ayuda a entender el funcionamiento del código y a resolver problemas de manera más efectiva.

### 3. ¿Por qué es importante aprender a usar Git y GitHub?

Git y GitHub son herramientas fundamentales en el desarrollo de software, ya que permiten gestionar el control de versiones de los proyectos y facilitan el trabajo en equipo. Además,

muchas empresas los utilizan, por lo que es importante contar con habilidades en su uso.

#### **4. ¿Es recomendable aprender JavaScript antes de usar un framework?**

Sí, es recomendable tener un buen entendimiento de JavaScript antes de saltar a un framework como React o Vue. Comprender las bases del lenguaje ayuda a evitar problemas y facilita el trabajo con el código en proyectos tanto personales como profesionales.

#### **5. ¿Es suficiente lo que se aprende en la universidad para trabajar como desarrollador?**

No, no es suficiente. Si bien la universidad aporta conocimientos técnicos, es importante complementarlos con experiencias y proyectos por fuera de la institución. Las tecnologías y enfoques que se usan en la industria del desarrollo suelen ser diferentes a los que se enseñan en la universidad.

*Por favor, consulta con un profesional en el campo si tienes dudas específicas sobre tu situación.*

¡Gracias por leer y no olvides revisar nuestros artículos relacionados para obtener más información sobre programación y desarrollo de software!

Saludos,

Infocivica.org

