

Incluyendo consejos de los mejores desarrolladores  
Google, Microsoft, Spotify, Adobe y más

# 14 HABITOS DE DESARROLLADORES ALTAMENTE PRODUCTIVOS

**Zeno Rocha**

# Parte Uno: Principios

## Hola Mundo

*"No existe ascensor hacia el éxito, tienes que usar las escaleras." — Zig Ziglar*

Empecé a desarrollar software hace más de diez años. Desde entonces, he creado muchas aplicaciones, he creado docenas de proyectos de código abierto y he escrito miles de commits. Además de eso, he hablado en más de cien conferencias y he tenido la oportunidad de conversar con un montón de desarrolladores por el camino.

He tenido la suerte de estar en contacto con algunos de los mejores ingenieros de software de la industria, aunque también he conocido a muchos programadores que siguen haciendo lo mismo durante muchos años.

¿Qué separa a un grupo del otro? ¿Qué tiene de especial la gente que trabaja en las empresas más grandes de nuestra industria? ¿Qué tienen de especial las personas que crean las aplicaciones más utilizadas en el mundo? ¿Cómo pueden algunos desarrolladores ser tan prolíficos en el trabajo y también fuera de ellos?

Estas preguntas permanecieron en mi cabeza durante mucho tiempo. Me di cuenta de que podía comprar los mejores teclados mecánicos, ir a las conferencias tecnológicas más famosas del mundo y aprender todos los frameworks más nuevos. Aún así, si cultivase malos hábitos, sería imposible convertirme en un desarrollador de primer nivel. Por ello decidí contactar a los

mejores desarrolladores que conozco y pedirles consejos sobre cómo ser más productivos.

Este libro no ofrece un camino directo o una fórmula predefinida para el éxito. Este libro es el resultado de una búsqueda. Una búsqueda para descubrir qué hábitos se pueden cultivar para ayudar a convertirse en un ingeniero de software mejor.

## Metodología

Este no es un libro tradicional. No encontrarás el mismo formato o estructura que tiene un libro normal. De hecho, este libro fue diseñado para ser lo más simple y objetivo posible. Puedes seguir el orden de los capítulos, o puedes leerlos individualmente. Todo es independiente y no depende de conocimientos previos.

Al final de cada hábito encontrarás una sección marcada como "**Preguntas y respuestas**", donde entrevisto a desarrolladores senior y líderes tecnológicos de varias compañías para comprender cómo llegaron allí. Fui tras gigantes tecnológicos como Google, Amazon, Microsoft y Adobe. Potentes startups como GitHub, Spotify, Elastic, Segment, GoDaddy y Shopify. Todo el camino hasta organizaciones establecidas como Citibank, BlackBerry y The New York Times.

Estas personas provienen de todo el mundo y tienen una formación bastante diversa. De San Francisco a Nueva York. De São Paulo a Montreal. De Londres a Estocolmo. La idea es presentarte, no el punto de vista de una persona, sino una colección de ideas sobre cómo navegar en tu carrera.

También encontrarás secciones marcadas como "**TODO**", donde te animo a reflexionar sobre ciertos temas y tomar medidas con instrucciones específicas. Recomiendo que te tomes unos minutos para sumergirte en ellos, ya que esto generará aún más conocimiento.

Y finalmente, puedes encontrar algunas secciones marcadas como "**Bonus**". Estos son contenidos adicionales que preparé para ti. Se pueden encontrar fuera del libro y complementarán lo que estás leyendo.

## ¿Por qué hábitos?

Si alguna vez has tratado de perder peso, sabes lo frustrante que es todo el proceso. Puedes hacer ejercicio tan duro como puedas durante tres horas, pero si lo haces solo una vez por semana, no tendrá ningún efecto. Lo que realmente genera resultados es cuando lo realizas varias veces por semana. Luego, de repente, unos meses después, comenzarás a notar cambios en tu cuerpo.

La consistencia es importante, y ese mismo concepto se aplica también a tu carrera profesional. Las cosas llevan tiempo, y la intensidad no siempre es la respuesta. Los hábitos que decidas cultivar (o no cultivar) determinarán tus oportunidades de vida futuras. Tal y como se describe en el libro *Atomic Habits*:

*"Los hábitos son el interés compuesto de la superación personal. De la misma manera que el dinero se multiplica a través del interés compuesto, los efectos de sus hábitos se multiplican a medida que los repites. Parecen marcar poca diferencia en un día determinado y, sin embargo, el impacto que producen a lo largo de los meses y años puede ser enorme. Es sólo cuando se mira hacia atrás dos, cinco o quizás diez años después que el valor de los buenos hábitos y el coste de los malos se vuelven notablemente evidentes."*

— James Clear

A menudo recibo correos electrónicos de otros ingenieros de software preguntando qué lenguaje de programación deberían aprender. Y aunque esa es una pregunta muy importante, creo que una pregunta más valiosa sería: "¿Qué hábitos necesito cultivar para ser eficaz en cualquier lenguaje de programación?"

Es por eso que decidí enfocar este libro en los hábitos en lugar de las tácticas.

¡Ahora vamos a sumergirnos! ¿Estamos preparados?

# Hábito 1: Busca las señales

*"Las palabras más cortas y antiguas, sí y no,  
son las que requieren más reflexión." - Pitágoras*

Todos los días somos bombardeados con tweets, newsletters y videos que nos dicen qué debemos hacer, qué debemos aprender y en qué debemos centrarnos. Nos enfrentamos constantemente con FOMO (*Fear Of Missing Out*), o "el miedo a perderse algo". ¿Qué pasa si estamos perdiendo el tiempo con el lenguaje de programación incorrecto? ¿Qué sucede si no estás usando actualmente el framework más productivo?



Personalmente, para mí todo comenzó con la elección del sistema operativo (SO) correcto. En ese momento, tenía un equipo Windows, pero todos me decían que los equipos Apple eran mejores y que debía cambiar. Sin embargo, sus precios estaban fuera de mi alcance, por lo que acceder a uno ni siquiera era una posibilidad. Un par de años más tarde uno de mis empleadores me dio la opción de elegir entre Windows y MacOS. Escogí directamente MacOS para ver de qué se trataba y por qué eran tan increíbles. Después de algún tiempo

usando MacOS, Ubuntu comenzó a ser muy popular y todos me decían que debía cambiarme a Ubuntu. Así que pensé en probarlo también y comencé a usarlo. Mi apreciación aquí no es decirte cuál debes elegir, sino apuntar que **no existe la mejor herramienta**.

En cambio, deberíamos practicar JOMO (*Joy of Missing Out*). o "la alegría de perderse algo", que se trata principalmente de ser feliz y estar contento con lo que ya se sabe. Confía un poco más en ti y recuerda todo lo que has aprendido hasta ahora. Por supuesto, no debes ser complaciente y dejar de estudiar nuevas tecnologías. Sería mejor si encontrases un equilibrio entre practicar las habilidades que ya tienes, y aprender otras nuevas.



La gente intentará convencerte de cuál es el mejor sistema operativo, el mejor lenguaje de programación, o el mejor framework. Te informarán sobre todas las cosas increíbles que hace su herramienta y que tu herramienta no hace. Sin embargo, la realidad es que cada herramienta es diferente y nosotros también somos diferentes como usuarios. Lo que es mejor para ti puede no ser lo mejor para mí o para otros.

Piensa en ello como una estación de radio que estás tratando de sintonizar. Sé que probablemente ya no uses radios, pero quédate conmigo. Imagine que giras el dial y sólo está captando ruido estático, pero después de unos segundos frustrantes, finalmente logras captar una señal y sintonizar una estación. La señal es la información significativa que realmente te interesa. El ruido estático es solo una variación aleatoria no deseada que interfiere con la señal. Es por eso que la autoconciencia es tan importante que debes poder identificar cuál es la señal y qué es simplemente ruido.

Es crucial entender que el ruido siempre estará ahí. No necesitas necesariamente abandonar las redes sociales, darte de baja de todas las newsletters y dejar de ver videos en YouTube. Una desintoxicación digital por supuesto que puede ayudarte por un tiempo; sin embargo, no es una solución a largo plazo. Lo que debes hacer es elegir qué es relevante para ti en este momento de tu carrera.

Acepta el hecho de que simplemente no puedes aprender todo. **Recuerda, los deseos son infinitos; las necesidades son limitadas.** Acepta el hecho de que lo nuevo no siempre es mejor. Hay personas que trabajan con lenguajes de programación antiguos y aún ganan mucho dinero. Practica diariamente el sutil arte de decir 'no'. No a esa librería más nueva. No a esa plataforma más elegante. Di más "no" para poder decir "sí" a lo que realmente te importa.



**1 SÍ !== 1 NO**  
**1 SÍ=== MUCHOS NOES**

Escucha el ruido, pero presta atención a las señales.

**// TODO**

Crea una lista de todas las tecnologías y herramientas que te gustaría aprender. Ahora etiqueta cada uno de ellas con una prioridad diferente: "Esta semana", "El Próximo Mes", "El Próximo año". Siempre que sientas que te estás perdiendo una nueva tendencia brillante, vuelve a visitar esta lista y reorganiza la prioridad.

## # Preguntas & Respuestas: ¿Cómo decides qué tecnología aprender e invertir tiempo?

Daniel Buchner (Microsoft):

*"Al principio de mi carrera, recuerdo haber prestado mucha atención a cada nuevo framework que aparecería en la portada de Hacker News. Leía sobre cualquier enfoque que adoptaran de manera única, y pasaba ciclos adicionales jugando con el framework o la librería para hacerme una idea. Esto está perfectamente bien si tienes suficiente tiempo libre como para explorar y aprender, pero a menudo puede provocar fatiga, porque la lista de cosas nuevas nunca termina.*

*Actualmente, trato de concentrarme en algunas cosas clave que impulsan mis evaluaciones:*

- 1. ¿Cuáles son los requisitos técnicos absolutamente imprescindibles para lo que sea en que esté trabajando? Esto puede incluir cosas como el rendimiento, la experiencia de usuario deseada o la interoperabilidad con otros sistemas.*
- 2. ¿Cómo de fácil será para otros trabajar en lo que estoy construyendo, y cómo será para ellos integrarlo en cualquier proyecto en el que estén trabajando?*
- 3. ¿Está alineado lo que estoy haciendo con la trayectoria de la web abierta, los estándares y las especificaciones que creo que se mantendrán a largo plazo?*

*Los tres puntos anteriores tienden a eliminar muchas de las librerías, frameworks y otras utilidades que encuentro. Esto me ha ahorrado tiempo y*

*me ha permitido centrarme más en ofrecer lo que necesito, en lugar de quedar atrapado en el turismo de desarrollo."*

Addy Osmani (Google):

*"Acepta que no puedes aprender todo, pero puedes aprender lo suficiente como para ser efectivo. Cuando parece que una idea, un framework o una tecnología están ganando aceptación, invierto una tarde en probarlo yo mismo para tener una idea de dos cosas: "¿Mejora mi productividad?" y "¿Mejora la experiencia del usuario en los tipos de proyectos que suelo construir?" Si la respuesta a cualquiera de estas preguntas es afirmativa, consideraré pasar más tiempo aprendiendo sobre el framework o la tecnología con más profundidad.*

*Debido a que nuestro tiempo es finito, hay muchas tecnologías que probaré, o que las encontraré lo suficientemente atractivas como para echarles un ojo, pero también tendré que valorar si aprender e invertir en otra cosa. Creo que esto es saludable. Originalmente probé (y aposté por) React el primer año que salió, ahora lo uso con regularidad. Me gustan muchas de las ideas en Svelte, pero como ya había aprendido React, Preact y Vue, decidí que era mejor invertir mi tiempo subiendo de nivel en áreas completamente diferentes en ese momento, por ejemplo la API de animaciones web.*

*Como dije, siendo el tiempo limitado, es bueno encontrar un equilibrio en lo que eliges aprender para mantenerte efectivo. Cuando eliges solo unas pocas opciones con un alto impacto, te preparas para tener tiempo suficiente para profundizar más en la tecnología elegida. Esto puede ser inmensamente útil cuando intentas construir algo no trivial en el mundo real."*

Gracias por tomarse el tiempo de leer este capítulo gratuito.

Espero que quieras leer el libro completo ahora :)

Quieres ayudar? Comparte con tus amigos!

- Zeno