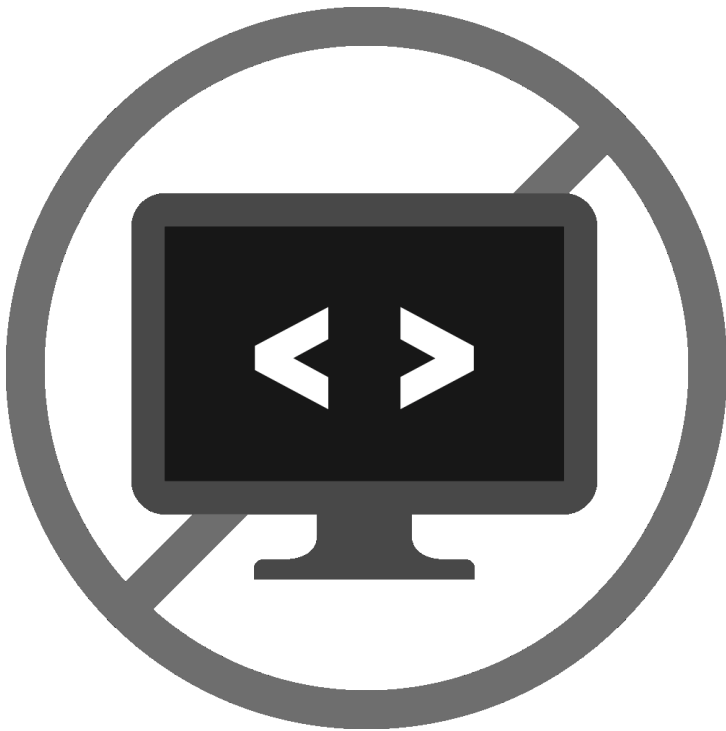


El Libro Negro del Programador_



El Libro Negro del Programador_

Cómo conseguir una carrera de éxito desarrollando software y
cómo evitar los errores habituales

Rafael Gómez Blanes



¿TE HAS SENTIDO ASÍ ALGUNA VEZ?

El Libro Negro del Programador te va a mostrar qué distingue a un buen profesional del software de otro que «juega» con la tecnología sin conseguir resultados importantes: en el nuevo paradigma laboral trabajaremos principalmente con la máxima productividad y por resultados, para lo que es imprescindible cultivar buenos y productivos hábitos de trabajo.

El autor_ |

Rafael Gómez Blanes es Ingeniero Informático por la Universidad de Sevilla (España). Infoemprendedor, ha trabajado en proyectos software internacionales relacionados con el sector eléctrico. Desarrollador profesional desde el año 1998, es experto en *clean code* y todas aquellas prácticas metodológicas que incrementan la productividad, mejorando la calidad del software generado. Evangelista de software ágil, dirige actualmente un equipo de desarrollo en una compañía de ingeniería realizando productos para la gestión de *smart meters* y su despliegue en la nube en modo SaaS (*software as a service*).

Introducción

«Emplea tu tiempo cultivándote a través de los escritos de otros, así ganarás fácilmente lo que para nosotros ha sido una dura tarea.»

(Sócrates)

Estamos viviendo momentos apasionantes: un nuevo paradigma económico ha venido para quedarse, podemos contratar casi cualquier servicio en cualquier parte del mundo, podemos diseñar aquí y fabricar por encargo a miles de kilómetros de distancia y del mismo modo lo que fabricas y produces en tu compañía puede ser comercializado en muchas partes del mundo. Profesiones que antes no existían ahora están a la orden del día, una rotación laboral alta está dejando obsoletos los puestos de trabajo «para toda la vida»; hay quien define todo este escenario como crisis, otros como un paradigma y orden nuevo cargado de oportunidades. Hay opiniones entre las que elegir.

Sin embargo, todo esto ha sido posible gracias al despliegue global de redes de comunicaciones que nos presentan a «golpe de clic» oportunidades que antes eran inimaginables. Hay quien dice que todo lo que estamos viviendo es una consecuencia natural de la implantación y uso de Internet.

El protagonista de esta revolución silenciosa, es, en esencia, el desarrollo de software: todo a nuestro alrededor está informatizado y, sin embargo, la profesión informática sufre de cierta incompreensión por el resto de la sociedad: a los desarrolladores de software se les encasilla fácilmente, se les asocia frívolamente con una imagen de «juventud», de trabajo en garajes destartalados cuando la realidad es que un buen proyecto sólo sale adelante con un alto grado de especialización y experiencia.

¿Somos conscientes de la enorme facilidad que tenemos hoy para emprender y poner en marcha ideas innovadoras desde el salón de casa?, ¿de que las barreras de entrada en muchos negocios han caído estrepitosamente? ¿Nos damos cuenta de que toda esta revolución la está protagonizando de una u otra manera el software que se ejecuta en millones de servidores por todo el mundo?

Recientemente se ha publicado un estudio según el cual en EEUU ya es mayor la masa laboral que trabaja «desarrollando aplicaciones» que el número de trabajadores dedicados a la agricultura, sorprendente, ¿no? Igualmente, no paro de leer, en estos mismos días, que el desarrollo de software será una de las profesiones más demandadas en el futuro próximo, lo está siendo ya, pero esta vez en forma de analistas web, SEO (*search engine optimization*), expertos en *big data*, *community manager*, etc. Es una profesión de futuro, efectivamente (de presente lo es ya), pero me pregunto cuáles serán las características que distinguirán a un buen profesional de otros cuyo trabajo deja que desear.

No obstante, la realidad es a veces totalmente frustrante: el poder de la tecnología se desvanecería si no tuviera el soporte de miles de desarrolladores de software afanándose día a día en que esta no colapse, haciendo parches (*hotfixes*) de soluciones cutres, manteniendo webs horribles cuyas compañías quieren rentabilizar al máximo o desarrollando *apps* para obtener algunos ingresos pasivos... Es una profesión poliédrica, multidisciplinar en la que dentro de la misma te puedes dedicar a hacer multitud de cosas extraordinariamente diferentes. Pero todas, absolutamente todas, se hacen bien, con profesionalidad, si somos capaces de rodearnos de buenos hábitos y de comprender la naturaleza frágil, artística y creativa del software.

Y luego viene la cadena de mando..., cuando el buen hacer técnico, los años de experiencia desarrollando arquitecturas software eficientes y correctas se tiran a la basura porque la única manera de prosperar (ganar más dinero, lógicamente), es pasar «al siguiente nivel» como manager y comenzar a trabajar de lo que un informático desarrollador de software apenas tiene idea, para lo que apenas o nada ha sido formado académicamente: gestionar un equipo de trabajo. ¿Pero qué tendrá que ver poner en marcha una arquitectura SOA o una metodología *extreme* con gestionar el tiempo y el trabajo de un equipo de personas, las cuales en su mayoría te miran con aprensión porque ven que para ellos cobrar algo más es hacer lo que tú acabas de conseguir?

En estos lamentables casos, la mentalidad empresarial dominante que te premia con asumir más «responsabilidad» lo que consigue es perder un magnífico técnico y ganar un pésimo gestor. A mí esto me ha pasado y es comidilla habitual en el sector. Y, sin embargo, ¡cómo desprestigia esta dinámica e inercia el hecho de programar!, ya que, si te pagan más por «gestionar y coordinar» será que lo otro es menos importante... ¿Conocéis quien tenga mejor retribución económica programando que gestionando grupos de trabajo o proyectos?

Entonces, ¿qué futuro laboral se puede esperar de un amante de la tecnología, de un apasionado empleado que pasa tantísimas horas delante de un ordenador depurando ese error sutil durante horas o de quien se esfuerza por generar el mejor software mantenible y ligero pero que ni su jefe ni su cliente van a valorar porque a estos lo único que les interesa medir es el tiempo y el coste a corto plazo (cosa lógica, por otra parte)?

El desarrollo de software es una profesión de presente y de enorme futuro, y, sin embargo, no sabemos aprovechar al máximo este nuevo paradigma económico cargado de oportunidades: caemos siempre en los mismos errores, una y

otra vez, sin darnos cuenta de que la única ventaja competitiva que tendremos será la de la calidad e innovación.

Por eso os presento este libro, donde resumo mi más o menos dilatada experiencia laboral y de *freelance*, llena de desaciertos, errores y decepciones pero también llena de éxitos profesionales que me han permitido seguir amando mi actividad diaria. En El Libro Negro del Programador encontrarás los errores más habituales que solemos cometer los desarrolladores profesionales de software y, sobre todo, veremos cómo evitarlos. Lo más interesante de esto es que la mayoría de esos errores, tropiezos y problemas que terminan haciendo fracasar un proyecto, no son de carácter técnico.

A diferencia de otras profesiones, el factor vocacional en el desarrollo de software es importante: los mejores desarrolladores son los que verdaderamente disfrutan más con lo que hacen. Y es que, ¿no deberíamos aspirar a amar todo lo que hacemos?, ¿o es que nos tenemos que conformar con pasar ocho o más horas al día vendiendo nuestro tiempo para pagar las facturas? Vamos hacia una economía en la que el tiempo que pasamos en un trabajo será más y más irrelevante: lo más importante serán los resultados y objetivos cumplidos; nos pagarán por proyectos y el trabajo «en nómina» será cada vez más escaso, sólo para el personal estructural necesario para la supervivencia de la compañía.

Aquí va el apasionante viaje de un ingeniero informático obsesionado por mostrar la especial idiosincrasia de nuestra profesión y, sobre todo, enseñar que hay muchas circunstancias no técnicas que «rodean» nuestra actividad y que determinan nuestro éxito o fracaso.

Quiero mostrar también en qué términos se debe considerar un software bien hecho como parte de un proceso más «artístico» que técnico; para los que aún no os habéis dado cuenta, los desarrolladores de software tenemos más de artistas

que de frías mentes matemáticas. Creamos artefactos de la nada que tienen que evolucionar y ser mantenidos durante mucho tiempo.

Simplicidad, diseños emergentes, deuda técnica, los hábitos del buen desarrollador, trabajar en equipo, refactorings, desarrollo de software testeable, la rentabilidad metodológica, horas o productividad, buena organización, abstracción bajo principios, disciplina de trabajo, reutilización y desacoplamiento de módulos y librerías, (y un larguísimo etcétera) son los ingredientes y el vocabulario habitual de este libro con el que espero crear, formar y motivar a más y mejores profesionales.

Este libro está escrito por un desarrollador de software por y para otros programadores y desarrolladores. Creo firmemente que el mejor currículum que podemos mostrar es el de un buen trabajo realizado con la máxima calidad y sobre todo de enorme utilidad. De nosotros depende, en cierta medida, rodearnos de las «condiciones» adecuadas para conseguir ese objetivo.

Índice de capítulos_ |

El autor

Introducción

Excepciones lingüísticas

Manifiesto

Desarrollo, pruebas, refactoring(todo va en el mismo paquete)

Qué es tener éxito en un proyecto software

Todo es cuestión de principios

Una vuelta de tuerca a la ley del cambio

Atreverse a eliminar lo implementado

Cuando incorporar más gente es desastre asegurado

Cuando el gestor de un proyecto es su mayor enemigo

El día a día de un buen desarrollador

El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

El mal entendido rol de arquitecto de software

La rentabilidad metodológica

Todo está en la UI

Diletantismo tecnológico

No se trata de trabajar más horas, sino de trabajar mejor

Sobre frameworks, librerías y cómo reinventamos la rueda

Los buenos desarrolladores escriben código depurable

Esclavo de tu propia solución o cómo querer ser imprescindible

Aprendemos de otros(o cómo no ser un desarrollador perezoso)

Potenciando la productividad con la gestión de la configuración e integración continua

La era del emprendimiento

Conclusiones

El test del desarrollador de software altamente productivo

Bibliografía

Referencias