

9 de febrero de 2017
Bogotá D.C.



William Sierra	william.sierra@exsis.com.co
Carlos Cardona	carlos.cardona@exsis.com.co
Ervid Molina	ervid.molina@exsis.com.co

The Clean Coder Capítulo 6: Practicando.

Todas las personas practican sus habilidades cuando se necesita mejorar el desempeño en su profesión. Por eso analizaremos cómo un desarrollador puede practicar su arte.

¿Cuántas veces un desarrollador ha explorado un nuevo lenguaje a través del popular “Hello World!”? Demasiadas, porque es una manera de probar que podemos escribir y ejecutar cualquier sentencia de código en ese nuevo lenguaje. La verdad es que los programadores generalmente no se toman la tarea de realizar pequeños programas para realizar sus pruebas, debido a que están muy enfocados en escribir código, para dedicarse a pensar en pulir sus habilidades. Además, solo hasta hace unos años se vio la necesidad de tomar decisiones rápidas en la industria del desarrollo de software.

Analicemos qué ha cambiado con el paso del tiempo. Algunas cosas han cambiado bastante, otras no tanto. Por ejemplo no podemos comparar el hardware de hace 40 años con el actual, pero seguimos escribiendo sentencias IF, ciclos WHILE y asignaciones. Es cierto que ahora hay mejores herramientas y más alternativas para escribir esas sentencias. Pero la naturaleza de las sentencias (que son nuestra materia prima) no ha cambiado.

Podemos ver que hace muchos años una compilación de un bloque de código se demoraba bastante. Hoy en día difícilmente se puede ver esa situación. La compilación toma un tiempo apreciable y las pruebas parciales toman algunos segundos. Algunas veces es necesario tomarse un tiempo para generar un código y otras veces se requiere generarlo en la menor cantidad de tiempo.

Hacer cualquier cosa rápidamente requiere práctica. Estar cambiando entre codificación y pruebas constantemente, requiere una toma de decisiones muy velozes. Tomar decisiones rápidamente significa ser capaz de reconocer un gran número de situaciones y problemas, para luego simplemente saber qué hacer para abordarlos.

Cuando un desarrollador está cambiando entre codificación y pruebas de forma continua y rápida, es porque su cuerpo sabe que teclas presionar. Una parte primitiva de su mente reconoce la situación y reacciona en milisegundos con la solución apropiada mientras su mente se concentra en un problema que requiere mayor análisis. Caso similar al de las artes marciales, donde la velocidad depende de la práctica. Y en ambos casos la práctica es similar. Elegimos un repertorio de pares (problema / solución) y los ejecutamos una y otra vez hasta que los recitamos de memoria.

Ahora pensemos en el dojo de la programación (Dojo es el término empleado en Japón para designar un espacio destinado a la práctica y enseñanza de la meditación y/o las artes marciales tradicionales o modernas.). Hacemos la analogía cuando un grupo de desarrolladores se reúnen y practican juntos así como los marcialistas. En otras ocasiones, los desarrolladores practicarán solos.

Hay diferentes prácticas que se hacen en un dojo, y que se pueden comparar al campo del desarrollo. Éstas son algunas:

KATA: Conjunto preciso de movimientos sincronizados que representan un combate. El objetivo, que se aproxima asintóticamente, es la perfección.

El propósito de aprender una kata no es realizarlo en el escenario. El objetivo es entrenar a su mente y cuerpo sobre cómo reaccionar en una situación de combate particular. La finalidad es hacer que los movimientos perfeccionados sean automáticos e instintivos para que estén allí cuando los necesite.

Una kata de programación es un conjunto preciso de pulsaciones de teclado coreografiadas y movimientos de ratón que simulan la resolución de algún problema de programación. En realidad, no está solucionando el problema porque ya se conoce la solución. Por el contrario, están practicando los movimientos y las decisiones involucradas en la solución del problema. La asíntota de la perfección es una vez más el objetivo.

WASA: Es muy parecido a una kata de dos hombres. Las rutinas se memorizan y se reproducen con precisión. Un participante desempeña el papel del agresor, y el otro el de defensor. Los desarrolladores realizan movimientos que se repiten una y otra vez (como el ping pong). Los dos participantes eligen una kata, o un problema simple. Un desarrollador escribe una prueba de unidad, y luego el otro debe hacer que su código supere la prueba. Luego se invierten los roles.

RANDORI: Combate de forma libre. Entendida en el desarrollo como la reacción a cambios repentinos en sesiones de práctica tanto en la fase de pruebas como de desarrollo.

Finalmente reflexionemos sobre el abordaje de la experiencia. No es saludable para la profesión que un desarrollador no diversifique las soluciones que da a los problemas. Se han visto casos donde algunas empresas trabajan sobre un único lenguaje, una única plataforma y un único dominio. Como consecuencia encontraremos un currículo estrecho, y un miedo al cambio en procesos y tecnologías, que de ocurrir, los tomarán desprevenidos.

Una alternativa es hacer lo que hacen los abogados y los médicos: contribuir a un proyecto de código abierto. Hay muchas propuestas libres, y probablemente no hay mejor manera de aumentar su conjunto de habilidades que realmente trabajar en algo importante de alguien más.

Los desarrolladores profesionales practican en su tiempo personal. No es el trabajo de su empleador ayudarle a mantener sus habilidades refinadas para usted, ni de mantener su currículo actualizado.

Por ello, concluimos que de una forma u otra, todos los desarrolladores profesionales realizan prácticas. Lo hacen porque se preocupan por hacer el mejor trabajo posible. Es más, practican en su tiempo personal porque se dan cuenta de que es su responsabilidad -y no la de su empleador- mantener sus habilidades actualizadas, para conseguir un beneficio para sí mismo y para el empleador.