

31 de enero de 2017
Bogotá D.C.



William Sierra	william.sierra@exsis.com.co
Carlos Cardona	carlos.cardona@exsis.com.co
Ervud Molina	ervud.molina@exsis.com.co

The Clean Coder Capítulo 4: Codificando.

En esta sección del libro se busca analizar la codificación así como su contexto. Para ello, es importante detectar y minimizar los errores durante este proceso, para que no se vuelvan a cometer incrementando su autoconfianza. Se detallarán diferentes principios morales, mentales y emocionales que se deben tener en cuenta para codificar.

Primero analizaremos las características que todo código y su desarrollador deberían cumplir. Es una actividad intelectualmente desafiante y agotadora. Requiere un nivel de concentración y enfoque que otras pocas disciplinas requieren. Encontramos las siguientes particularidades:

- Funcionamiento: Debe asegurarse de entender el problema, con el fin de que el código que usted escribe es una representación fiel de la mejor solución posible.
- Resolución del problema establecido por el cliente: Los requisitos del cliente no solucionan realmente los problemas. Por tanto, debemos encontrar y extraer lo que realmente necesita el cliente.
- Acoplamiento al sistema existente: No debería aumentar la rigidez, fragilidad u opacidad del sistema.
- Legibilidad para otros desarrolladores: Dicho código debe reflejar la intención que tiene (no basta solo con comentarios).

Ahora reflexionemos sobre las condiciones que se deberían considerar para poder codificar.

La primera es no codificar cuando esté cansado. Para alcanzar el profesionalismo se necesita más disciplina que horas. Asegúrese de dormir, de su salud y un estilo de vida donde enfunde 8 buenas horas laborales. De lo contrario el resultado no será el deseado. Se recomienda encontrar una manera de eliminar las distracciones y despejar su mente.

La segunda es sobre las preocupaciones. Es muy difícil concentrarse en codificar si personalmente usted no dispone la mente para esa actividad, ya que por detrás usted está dando prioridad a resolver otros percances. Lo mejor es dedicar un tiempo a saber cómo se están resolviendo esas situaciones, para reducir la ansiedad. Por tanto, un desarrollador profesional asigna su tiempo personal para garantizar que el tiempo empleado en la oficina sea tan productivo como sea posible; además debería reservar tiempo en la casa para resolver sus ansiedades de modo que no las traiga a la oficina.

En tercera instancia encontramos la zona de “flujo”. Definida como un estado meditativo suave en el que ciertas facultades racionales disminuyen a favor de una sensación de velocidad, donde usted se enfoca en una única cosa. El problema de caer en esta “zona” es que la sensación de efectividad es temporal, no es realmente hiper productiva, no es infalible y se pierde la perspectiva general del proyecto. Se puede llegar a escribir mayor cantidad de código, pero usualmente este código requiere modificaciones más adelante: esto hace que se tenga que retroceder en el código y se tenga que rehacer por completo o modificar gran cantidad de él.

En cuarto lugar encontramos las horas extra. Son necesarias y funcionan en algunos casos, pero son directamente proporcionales al impacto que aporte al trabajo realizado. Se debe tener en cuenta que un desarrollador extenuado es más propenso a generar código defectuoso y a cometer más errores. Si su jefe inmediato no tiene preparada una alternativa en caso tal que las horas extra no logren su cometido, entonces éstas no sirven.

En quinta instancia podemos encontrar las falsas entregas. Estos casos se presentan cuando los desarrolladores dicen que la entrega está lista y realmente no lo está. Es una falacia muy costosa para la organización, porque se construyen falsas expectativas por creer que en un futuro cuando se tenga tiempo, se podrán resolver los errores acumulados. Si el gerente escucha que todo está bien, los informes de estado mostrarán que se están cumpliendo las fechas, desconociendo que al final el proyecto terminará en desastre. Una entrega estará lista cuando las pruebas y los analistas de negocio así lo indiquen.

En sexto lugar podemos hablar de la esperanza. Es un defecto crítico creer que un proyecto se puede entregar en una fecha estipulada cuando con argumentos se prueba que esa fecha es imposible de alcanzar. Esa vaga esperanza destruye cronogramas y arruina imágenes corporativas. Por ello se deben construir varias formas de adaptar un proyecto, disponiendo de alternativas económicas y oportunas; de modo que sin importar el camino que se escoja igual se alcanzará el objetivo. Así se evita dejar todo para última hora. Como complemento cabe agregar que apresurarse a cumplir una fecha va en contra del profesionalismo; se debe tener en cuenta las estimaciones hechas desde un inicio, si se requiere terminar en un tiempo anterior a las estimaciones, se deben reducir los alcances del proyecto para poder responder.

Transversalmente se encuentra la ayuda. Es importante solicitar ayuda y ofrecerla a los demás, pero esto implica que también hay que saber cómo se ofrece y como se pide. Rehusarse a responder los requerimientos de otros es una violación de la ética profesional, es responsabilidad de los desarrolladores prestarse ayuda entre ellos. Esto no significa que no se necesite tiempo a solas, lo que se puede hacer es establecer horarios en los que se esté disponible y otros en los que no. En determinado momento se va a necesitar ayuda; cuando alguien se ofrezca, se debe ser amable.

Otra forma de ayuda es la mentoría. Es cuestión de ética profesional que los desarrolladores más experimentados pasen tiempo con los desarrolladores más jóvenes a su cargo. Del mismo modo los desarrolladores más jóvenes tienen el deber profesional de buscar tutoría de los desarrolladores más experimentados.

Finalmente encontramos una serie de eventualidades, que pueden influir positivamente o negativamente en la codificación. Por ejemplo escuchar música, leer, ir a cine, ver televisión, entre otros; son actividades que nos ayudan a cultivar la creatividad para salir de la ansiedad que se puede producir en el trabajo. Estas medidas buscan precisamente que no se bloquee cuando codifique, producidas generalmente por falta de sueño, miedo y depresión. Cuando usted tenga sueño, esté cansado o se encuentre estancado, lo mejor es alejarse por un momento. Seguir trabajando en el mismo problema cuando se está cansado o estancado solo hace que se retrase más en encontrar una solución: lo mejor es dejar que nuestro subconsciente trabaje en la solución dándole algunas entradas creativas a nuestra mente.

Concluimos que el desarrollo de software es una maratón, no un sprint. No se puede ganar la carrera por tratar de correr tan rápido como se pueda desde el principio. Se gana conservando sus recursos y llevando el ritmo. Los desarrolladores profesionales conservan su energía y creatividad con el mismo cuidado. Además, un par desarrollador ayuda a salir del bloqueo, ya que se genera un cambio fisiológico cuando se trabaja en equipo. Aceptar la ayuda con gratitud, y seguir las sugerencias de los desarrolladores experimentados es de profesionales.