La ética del informático: "Un día haremos una estupidez que costará miles de vidas"

Cada vez más desarrolladores se atreven a hacer públicas las peticiones ilegales o escasamente éticas que les hacen sus jefes o clientes. Conforme avanza la automatización de nuestro entorno, el debate sobre la necesidad de un código deontológico para los programadores también se intensifica. En España, pocas carreras relacionadas incluyen una formación ética consistente.

Marta Sofía Ruiz (El diario.es)

21/12/2016 - 15:29h

Los errores de un informático pueden ocasionar miles de muertes

Casi todo lo que hacemos hoy en día —desde comprar o realizar una llamada hasta coger un tren, conducir un coche conectado o viajar en avión— implica utilizar algún **programa informático**. Detrás de las líneas de código que rigen nuestro día a día está el trabajo de un profesional que escribe las instrucciones y las reglas lo mejor que sabe, puede y le dejan.

Como son humanos, los desarrolladores pueden **cometer errores o tomar decisiones equivocadas**, a veces con graves consecuencias. También puede suceder, como han confesado algunos programadores arrepentidos o escandalizados por las peticiones de sus jefes, que el 'software' en el que se ven forzados a trabajar tenga unos **fines cuestionables**.

Así, mientras estos profesionales van escribiendo el código que mueve el mundo (sistemas de voto electrónico, herramientas bursátiles, complejos programas de gestión industrial...), el debate sobre la **necesidad de establecer códigos éticos en informática** y de dar directrices claras a los jóvenes estudiantes que se inician en el sector se intensifica, sobre todo en el mundo anglosajón.

"Estamos matando gente", se lamentaba el programador Robert Martin en una reciente intervención. "Y un día alguno de nosotros va a cometer una estupidez y el resultado va a ser una catástrofe en la que mueran miles de personas".

Por desgracia no iba desencaminado. "El mal funcionamiento de programas informáticos puede provocar graves problemas que incluso involucren el coste de vidas humanas", recuerda a HojaDeRouter.com José Manuel García Carrasco, catedrático de la Universidad de Murcia y experto en la materia. "Hay unas **prácticas de trabajo saludables que pueden contribuir a mitigar o resolver estos problemas** y ayudarnos a tratar el tema de la responsabilidad ante un fallo informático".

En los años 90, este investigador, miembro de la Red de Excelencia Europea HiPEAC y de prestigiosas asociaciones internacionales como IEEE y ACM, ya publicaba artículos hablando sobre la importancia de la ética para los programadores y **reclamando la elaboración de códigos deontológicos rigurosos**. Aunque entonces era difícil prever la omnipresencia de la tecnología en nuestras vidas, ya se habían producido algunos incidentes graves por culpa de problemas informáticos sobre los que convenía reflexionar.

"Algunos recordarán el caso del **buque de guerra inglés Sheffield**, hundido durante la guerra de Las Malvinas en 1982", apunta el catedrático. Aquel barco estaba entre los más modernos de su tiempo, y los pilotos argentinos que intentaban atacarlo contaban con aviones menos sofisticados. Sin embargo, fue alcanzado por un misil.

"Aunque la versión más divulgada fue que gracias a la pericia y valentía del piloto había conseguido hundir dicho barco, un examen más profundo del caso reveló que **el 'software' de defensa, que no había sido verificado en su totalidad, tenía algunos fallos**", explica Carrasco.

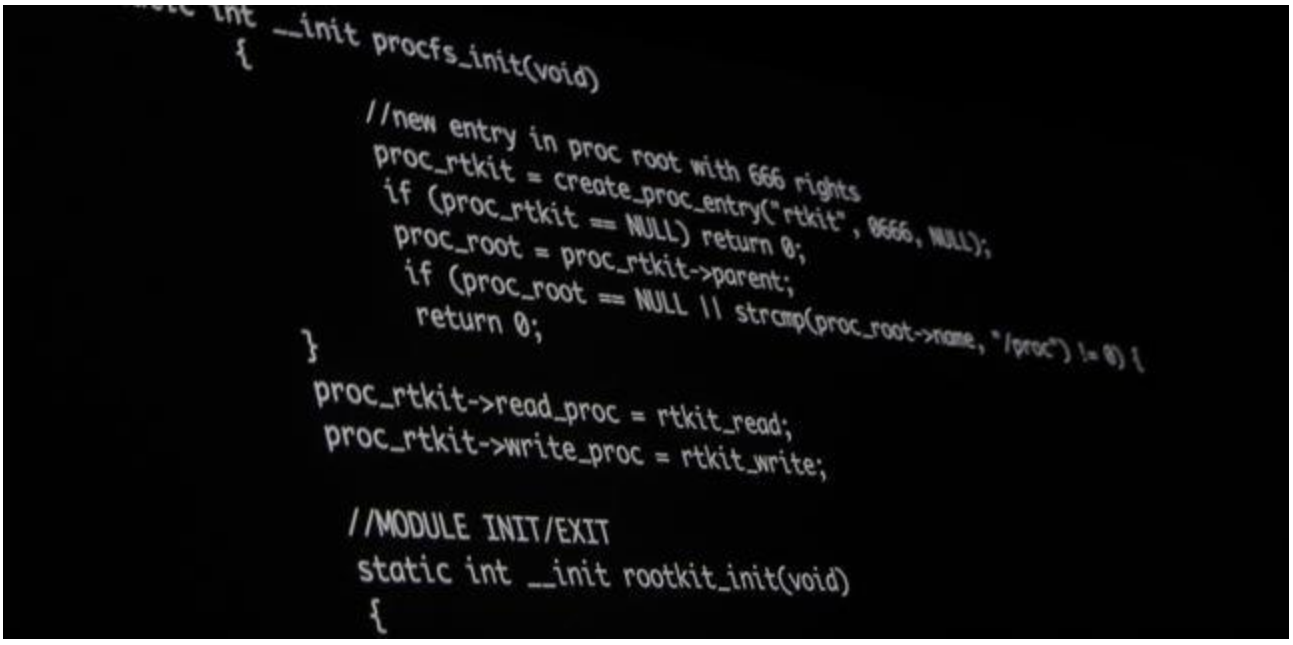
En aquellas décadas, otros conflictos bélicos dejaron más ejemplos de fallos informáticos que terminaron costando vidas. Tal y como relata el catedrático, en la primera guerra del Golfo, en el año 1991, **un misil iraquí del tipo Scud traspasó la barrera de defensa de misiles estadounidenses Patriot**, penetrando en una base de Estados Unidos en Dhahran (Arabia Saudí) y matando a 28 personas. La causa fue otro fallo en el 'software' de defensa.

Sin embargo, no todos los lamentables incidentes han tenido lugar en el contexto de una guerra. A finales de los años 80, el **equipo Therac-25 para el tratamiento del cáncer**, que se basaba en una terapia por bombardeo de rayos en la zona afectada, se hizo especialmente popular. Este aparato disponía de dos tipos de radiaciones: directas de baja potencia y reflejadas de alta potencia.

"Por un fallo en el diseño de la aplicación, en algunas circunstancias el equipo no operaba correctamente y le aplicaba al enfermo directamente las radiaciones de alta potencia, lo que **provocó que murieran varias personas** antes de que se detectara dicha anomalía y se retirara el equipo", explica Carrasco.

Desde entonces, aunque muy poco a poco, las formulaciones éticas han ido avanzando con el objetivo de evitar fallos como aquellos en un mundo cada vez más informatizado. "La importancia de este tema hoy en día es tan grande que ha hecho que ya se hayan desarrollado códigos de ética para los profesionales de la informática, especialmente en el terreno de la programación", concreta el experto.

Las organizaciones internacionales más prestigiosas, como la Association for Computing Machinery (ACM), el Institute of Electrical and Electronics Engineers (IEEE) o la International Federation for Information Processing (IFIP), han ido desarrollando códigos y normas de conducta aplicables a este sector. De hecho, en el año 1999, las dos primeras suscribieron un código ético para la enseñanza y la práctica profesional de los ingenieros de 'software', que resume los deberes de un programador en ocho principios, comenzando por **el interés público**.



¿Quién es programador? Los expertos se preguntan si debería determinarlo un colegio profesional

“Dicho código, que es el que está actualmente en vigor [...] vino a cubrir una necesidad muy importante”, afirma Carrasco. Sin embargo, esta referencia deontológica, que se elaboró cuando muchas de las tecnologías actuales ni siquiera existían, presenta ciertos problemas como, en algunos casos, la falta de concreción. Otro, muy actual y el principal según el catedrático, tiene que ver con **quién puede ser considerado un programador**.

“Para ser programador, ¿hay que haber estudiado Ingeniería Informática o se trata de un conocimiento de tipo artesanal que cualquiera, con pericia y tiempo, puede desarrollar?”, se pregunta el experto. “Dicho de otra forma: ¿puede cualquiera desarrollar y vender una aplicación informática o se necesita pertenecer al **colegio profesional** de ingenieros informáticos?”.

El resto de carencias son comunes a los códigos deontológicos de casi cualquier profesión: cómo se define lo que es correcto y lo que no, **cómo se controla su cumplimiento o quién y cómo se encarga de sancionar** a aquellos que se saltan las normas.

A pesar de que existen ciertos códigos, todavía queda mucho por hacer. En las universidades españolas, en las carreras relacionadas con la informática, **es poco frecuente encontrar alguna asignatura en la que se aborden cuestiones éticas** y solo alguna materia avanzada de programación recoge algún epígrafe sobre el comportamiento deontológico de un ingeniero de ‘software’. Además, los problemas a los que se tienen que enfrentar los profesionales del sector aumentan cada día.

En España, las carreras informáticas incluyen poca o ninguna formación deontológica

Ejemplo de ello es la creación de aplicaciones informáticas complejas, desarrolladas por un equipo de personas que, además de especificar el problema y escribir el código, tendrán que asegurarse de que funcionan correctamente. “Habitualmente **es imposible que se lleguen a 'testear' completamente**, por lo que el comprador se tiene que conformar con que haya una alta probabilidad de que el programa no tenga ningún error”, se lamenta el catedrático. “**Y cuando falla, ¿de quién es la culpa?** ¿Del programador, del especificador del problema, del que le hizo las pruebas o del que lo instaló?”. Todavía no hay respuesta clara.

La **propiedad intelectual** —dirimir quién es el dueño de una aplicación, si es lícito copiar un programa o hasta cuándo tiene que dar soporte el creador de una herramienta—, las **cuestiones de privacidad** en el almacenamiento de datos o la legitimidad del **acceso a un servidor** —dirimir si es razonable acceder si no está expresamente permitido (como harían un 'hacker' ético para buscar fallos) o si un programador puede dejar puertas traseras en sus aplicaciones— **son solo algunos de los dilemas** que se plantean los profesionales de la informática y que a menudo son difíciles de resolver.

Incluso si conocen la respuesta, en ocasiones **ceden a la presión de sus jefes** y acaban escribiendo programas cuyos fines no son del todo legales o éticos. El código que empleaban los coches de Volkswagen para pasar las pruebas de emisiones contaminantes o el uso cuestionable de los datos personales que hacen algunas tecnológicas podrían ser buenos ejemplos.

"Para solucionar en parte los problemas anteriores, mi propuesta es **que en los estudios de informática se incluyan una o varias asignaturas de deontología** que preparen a los estudiantes para comprender la programación de aplicaciones informáticas como una profesión dentro del contexto de la sociedad", reclama Carrasco. "Los estudiantes necesitan desarrollar la capacidad de preguntarse acerca del impacto social de la informática". Aprender ética para programar el mundo.

Las imágenes de este artículo son propiedad, por orden de aparición, de WoCinTech Chat, Aleksandar Cocek, Christiaan Colen y hackNY.org