

Transcripción del Artículo hecho por Carlos Prego en Xataka (2016).

Durante su paso por las facultades los futuros ingenieros informáticos estudian matemáticas, lógica, fundamentos de física, estadística, sistemas operativos, procesadores de lenguaje, programación... Un extenso y rico currículo en el que a menudo la ética ocupa un lugar secundario, apenas una asignatura a lo largo de un plan de estudios amplio y exigente. Cuando abandonan las aulas y se dedican a la programación gran parte de ellos se encuentran sin embargo con que su labor va más allá de las cifras, objetivas, mesurables y frías. En su día a día se ven obligados a encarar dilemas morales, más peliagudos conforme el software gana espacio en la sociedad y asume funciones sensibles, como la gestión de bancos de datos privados o información estratégica.

Las situaciones con implicaciones morales son más comunes de lo que cabría pensar. Y con frecuencia alcanzan calados dignos de los diálogos socráticos o las disquisiciones de Kant. A principios de 2017 varios programadores compartían con Xataka algunos de los breches éticos con los que habían lidiado en sus empresas: el desarrollo de software para la mejora de armas que luego se usan con fines ofensivos, programas para empresas petroleras con un impacto directo en la degradación del medioambiente... O simplemente el diseño de webs de venta que “presionan” a los usuarios con una subida gradual de tarifas para que compren cuanto antes.

Uno de los ejemplos más recientes y que muestra la clase de conflictos a los que pueden dar lugar los dilemas éticos cuando colisionan los intereses de las empresas y los empleados lo brinda Google. A finales de 2018 trabajadores de la multinacional alzaron la voz de forma pública contra "Project Dragonfly", un servicio censurado de búsquedas para el mercado chino que podría usarse además para la vigilancia. En una carta divulgada por Medium, los empleados de Google muestran su recelo sobre el uso que podría darse a la herramienta y piden a los líderes de la compañía que cancelen el proyecto. No era la primera vez que lo hacían. Solo unos meses antes, en verano, un millar de trabajadores de Google ya habían rubricado una carta en contra de "Project Dragonfly".

Las críticas de los empleados de Google a la dirección de la compañía por el proyecto Dragonfly es un ejemplo del choque que puede darse entre los intereses de una empresa y la conciencia ética de sus desarrolladores

El alcance cada vez mayor del software implica también que cuando se comete un error sus consecuencias van más allá de un simple fallo de cálculo. En 1991, durante la primera Guerra del Golfo, un misil iraquí hizo saltar por los aires una base estadounidense en Dhahran (Arabia Saudí) y acabó con la vida de 28 personas. El impacto fue posible porque el misil burló la barrera de defensa de EEUU. La razón: un fallo en el software. Un reportaje publicado por Eldiario.es en 2016 recuerda otro caso que deja a las claras el alcance de la programación: en la década de 1980 una tara en el diseño de un novedoso equipo para el tratamiento del cáncer provocó que varios pacientes recibiesen radiaciones de alta potencia que no necesitaban. Consecuencia: hubo fallecidos.

Cuando en la coctelera se meten además cuestiones como la propiedad intelectual, la privacidad en el almacenamiento de datos o la legitimidad de acceso a un servidor, la necesidad de unas directrices éticas para los profesionales se vuelve si cabe más acuciante.

Visto el poder y alcance de la programación, la pregunta del millón es: ¿Deben los desarrolladores dar más peso a los códigos deontológicos de su profesión? ¿Debería la justicia y la administración pública regularlos —igual que hace con los abogados, médicos o periodistas— para que un programador pueda hacer una objeción de conciencia sin arriesgarse a perder su empleo? O rizando un poco el rizo... ¿Deberían los desarrolladores de software salir de las facultades habiéndose comprometido con un juramento hipocrático, igual que los profesionales de la salud?

Códigos éticos internacionales.

A lo largo de las últimas décadas, organizaciones de prestigio internacional en el campo de la informática, como el Institute of Electrical and Electronics Engineers (IEEE), la Association for Computing Machinery (ACM) o la International Federation for Information Processing (IFIP) han elaborado códigos de conducta para sus propios profesionales. Hace dos décadas ACM e IEEE fueron un poco más allá y acordaron incluso un corpus para los ingenieros de software que destaca todavía hoy a nivel internacional como uno de los referentes deontológicos.

No son los únicos. Con frecuencia los colegios profesionales se dotan a sí mismos de pautas regladas. La iniciativa suele emanar de la toma de conciencia de los propios expertos, quienes

aprecian la responsabilidad de la labor que desempeñan y sus implicaciones crecientes. Los puntos en los que inciden son la integridad, transparencia, sostenibilidad o no colisionar con el interés público. Entre las entidades que se han dotado de pautas similares en España figuran, entre otros muchos, el Colegio Profesional de Ingenieros en Informática de Andalucía (CPIIAND), el de la Comunidad de Madrid (CPIICM), Catalunya (COEInf) o el del Principado de Asturias (COIIPA). A menudo recogen —como en el caso del CPIIAND— una relación de faltas e infracciones.

¿Son suficientes esas pautas profesionales?

Expertos en la materia advierten de los retos crecientes. Mary Eugenia Cartaya, licenciada en Ciencias de la Computación por la Universidad de La Habana, reconoce en su ensayo sobre ética y software el alcance cada vez mayor de los programadores, pero advierte de “riesgos insospechados que alcanzan a todas las esferas de la vida social”. “Es preciso que nos detengamos a reflexionar acerca de los valores éticos, porque están presentes en la primera relación de conexión entre el desarrollo tecnológico y la sociedad: el hombre”, señala: “Dada la repercusión de la informática en todas las esferas, cobra especial importancia la observancia de los valores éticos”.

El célebre Catedrático de Lógica y Filosofía de la Ciencia Fernando Broncano apunta en una dirección similar. “Hay que desarrollar códigos de buenas prácticas y si es posible algún tipo de juramento hipocrático en general en todas aquellas ingenierías que sean sensibles”, abunda. Broncano señala que “la iniciativa individual” de cada profesional juega un papel determinante, pero admite también que es en el punto en el que esta choca con las dinámicas y pautas de las empresas donde a menudo se pueden producir conflictos. Por muy concienciado que esté un desarrollador, ¿Cómo puede aplicar un código deontológico si su compañía no se lo permite?

Los expertos señalan que es el momento de abrir el debate ante el aumento de los ámbitos en los que se aplica el software: de la sanidad al transporte o las sugerencias de búsqueda, pasando por la venta online de billetes

"Me parece que es el momento para empezar a plantearse los límites éticos del software. La cuestión es si es algo que hay que cargar sobre las espaldas de los desarrolladores uno a uno

o si hay que plantearse algún tipo de código que obligue a las plataformas y empresas a introducir los códigos de auto restricción en lo que respecta a los límites éticos”, comenta Broncano, quien señala las ventajas de que los profesionales se doten de pautas que gocen de “respaldo social o institucional” y que las empresas dispongan de un “código ético de buenas prácticas”.

El reto no es sencillo. “Lo que ocurre con los abogados y médicos es que no solo manejan un código ético, sino una legislación positiva en la que se establecen límites al tipo de acciones que pueden realizar. En el caso de los abogados la confidencialidad, por ejemplo. Si hablamos de desarrolladores la cuestión es un poco más compleja. ¿Cuáles serían los contenidos de ese código ético, por ejemplo, en lo que respecta a si sus acciones van a tener efectos sobre el derecho a la privacidad de los usuarios? En casos así lo que puede hacer un escritor de códigos es una parte, pero la empresa tiene que hacer la otra, que es la más importante”, reflexiona el filósofo.

El debate se encuentra todavía en “una primera fase”, pero Broncano valora que al menos se haya puesto ya sobre la mesa y tomado conciencia de su relevancia. “Los límites éticos deben ser una consideración intrínseca desde el primer momento en el que te pones a programar”, señala el Catedrático de Filosofía de la Ciencia antes de abogar por una mayor formación ética en las aulas donde se forma a los desarrolladores de software. “Hay un déficit serio. En Medicina hay biomedicina, pero en las ingenierías no hay algo parecido”, apostilla.

¿Un juramento como el de los médicos?

¿Debería establecerse un juramento hipocrático para los programadores? Más allá del debate teórico, en los últimos años no han faltado intentos aislados por dotar a los desarrolladores de un juramento ético homólogo al de los médicos. En 2015 el célebre ingeniero de software estadounidense Robert Cecil Martin, más conocido como “Uncle Bob”, propuso una figura similar mediante la que los desarrolladores se comprometían a ejecutar su labor de la forma más eficaz, segura y honesta posible. En su best seller Armas de destrucción masiva, la matemática de EEUU Cathy O’Neil rescata la idea, inspirada en gran medida en The Financial Modelers Manifiesto, un documento lanzado en 2009 por los ingenieros Emmanuel Derman y Paul Wilmortt.

En EEUU incluso se organiza un evento, el Data For Good Exchange (D4GX), que pone énfasis precisamente en la necesidad de capacitar a los programadores para que encaren dilemas morales. En 2018, durante el D4GX celebrado en Nueva York, Bloomberg anunció una colaboración con Data Democracy y BrightHive, Inc. para dotar a los científicos de datos de un código ético y una serie de principios que les permitan afrontar los desafíos que supone su ejercicio profesional.

Las protestas de la plantilla de Google derivadas del proyecto Dragonfly o el escándalo de Cambridge Analytica en el que se vio envuelto Facebook no han hecho sino incidir en la urgencia de un código o pautas éticas para los desarrolladores de software. A pesar de ese escenario, los intentos por establecer un juramento hipocrático han sido tímidos y desde el sector aseguran que, al margen de que se formulen de forma aislada, no están extendidos ni reglados.

Broncano apuesta por ir paso a paso y, antes de plantearse la elaboración de cualquier juramento, ahondar en el debate ético. “Ahora mismo plantear un código hipocrático del tipo de Medicina sin haber planteado seriamente en qué aspectos consideramos que la programación excede los límites me parece que sería como poner la venda antes de la herida. Debemos aún examinar qué tipo de algoritmos deben ser sometidos al juicio del público”, reflexiona el filósofo.

No todas las voces coinciden en la utilidad o conveniencia de un juramento para desarrolladores. José Antonio Marín Casanova, profesor de Filosofía Actual, Tecnociencia y Sociedad en la Universidad de Sevilla, incide en que “el límite de la conducta de un programador, como el de cualquier ciudadano, es el código penal”. “El juramento hipocrático es un código ético sin validez legal. Otra cosa es que los programadores, como cualquier otro cuerpo profesional, cobren conciencia sobre las consecuencias sobre los demás de sus propias acciones”, previene.

No todos los expertos coinciden en la conveniencia de un juramento hipocrático y abogan por poner énfasis en la educación ética en los colegios y facultades

Para Marín Casanova la mejor salvaguarda de que los desarrolladores ejercerán su oficio de forma responsable es “una buena educación ética, pero también artística, literaria, histórica,

humanista, en definitiva. No solo para los programadores, sino para el ciudadano en general. Más allá de la concienciación, un juramento hipocrático sin eficacia jurídica solo tendría un valor simbólico”.

“Si queremos que los programadores, dado que trabajan con material muy sensible, no se propasen, es más conveniente la previsión legal que la moral. Y la ley, más bien, habría de proteger al programador con un reconocimiento análogo al del secreto profesional (reminiscencia del viejo secreto de confesión), para que no se vea obligado a revelar a nadie aquellos datos que exclusivamente ha conocido por el ejercicio de su oficio”, abunda el profesor de filosofía.

Pablo González, Security Researcher en Ideas Locas en el área CDO de Telefónica, reconoce la “complejidad” del debate y apunta las diferencias de calado que hay entre la labor que desempeña un programador y la de un médico o abogado. “Cuando un desarrollador se encuentra en un proyecto grande y complejo puede que no conozca su objetivo. Realiza una parte del sistema, un módulo o una parte de un pequeño módulo. No tiene por qué tener la visión del conjunto global, ni tampoco saber qué es lo que hará el sistema o la aplicación. Si partimos desde este punto, el disponer de un juramento hipocrático carece de sentido. Siempre habrá personas que sepan cuál es el objetivo del proyecto, pero no tiene por qué ser desarrolladores”, reflexiona.

“La situación entre los programadores y médicos no es igual. La forma en la que se trabaja y el conocimiento que cada uno tiene sobre el trabajo que realiza son diferentes. Un médico puede afrontar el problema concreto, mientras que el desarrollador solo puede conocer parte del problema, sin mayor conocimiento de la situación”, comenta González, quien valora que la responsabilidad de las implicaciones morales de los proyectos debe recaer sobre sus impulsores: “En los sistemas de IA de dudosa moral o los proyectos generales no éticos debería ser responsable la empresa o gobierno que hay detrás. El desarrollador puede no conocer el tipo de sistema que realiza”.

La dificultad de trasladar la figura del juramento hipocrático al campo de la programación no se limita solo a cómo trabajan los desarrolladores en las empresas. González revela otro punto igual de complejo: el intrusismo laboral, mucho menor en la abogacía o la sanidad que entre los desarrolladores de software. “El intrusismo aumenta y nos podríamos preguntar quién

debería tener el juramento hipocrático: ¿todos los que desarrollan? ¿Solo los colegiados? ¿Colegiados y con carrera? ¿O con módulo también valdría? Es algo complejo de acotar y en lo que resulta difícil ser justo. Además no tiene sentido en algo tan abstracto y donde el desarrollador carece de la visión global del proyecto que sí se tiene en otros sectores”, abunda el experto.

González reconoce en cualquier caso la importancia de un ejercicio profesional ético en el desarrollo de software. “Los programadores deben tener una ética adecuada y debe dársele mayor peso. Hay que tener en cuenta que actualmente y en ciertas situaciones es difícil marcar lo que es ético y lo que no. En algunos casos está muy claro, pero la inclusión de la tecnología en el día a día de la sociedad ha hecho que tengamos que revisar lo que es la ética y redefinir muchos casos”, apuesta.

Una tarea que arranca en los colegios.

El filósofo Aníbal Monasterio, autor del paper “Ética algorítmica”, apuesta por una mayor reflexión sobre “el impacto” de la programación. “Hoy la inteligencia artificial, la robótica y en general las ciencias de la computación penetran en todas las facetas de nuestra vida, desde la salud al transporte o los sistemas de recomendación. Todo esto tiene un impacto en la sociedad”, incide Monasterio, que al igual que Broncano apuesta por impartir más ética en las facultades.

“Al igual que en la Bioética se han establecido una serie de principios, como el de justicia, respeto o autonomía, las ciencias de la computación actualmente necesitan una serie de principios éticos que guíen su actividad”, anota. Aunque rehúye de los discursos relativistas o posmodernistas que sostienen que toda ciencia es construcción social, Monasterio reconoce que en su desarrollo sí se alcanza un componente social importante. “Los valores de la ingeniería y otras tecnologías tienen una importancia en el contexto social en el que se enmarca”, comenta.

Jesús Moreno León es Jefe del área de “Experimentos en el aula” del Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), organismo que depende del Ministerio de Educación. Aunque reconoce que las cuestiones sobre formación ética “en ocasiones no se tienen en cuenta como se debería” en las aulas, destaca la labor de la

administración para que los aspectos éticos de la programación estén presentes ya en ESO y Bachillerato.

Administraciones, empresas, colegios profesionales y universidades son cada vez más conscientes de la urgencia de abrir una reflexión sobre las implicaciones morales que pueden llegar a encarar los desarrolladores de software

A modo de ejemplo Moreno cita un reciente informe —publicado en octubre de 2018— que elaboraron conjuntamente el Ministerio de Educación, representantes de 14 gobiernos autonómicos, varias universidades y empresas líderes, como Apple, Microsoft, Everis o Telefónica, para ahondar en la presencia de la robótica, la programación y el pensamiento computacional en las aulas.

“Cada vez hay una sensibilidad más clara hacia este tema. En este grupo de trabajo ya hubo unanimidad a la hora de incluir este tipo de cuestiones entre los objetivos fundamentales que el alumnado debería tener. Creo que la situación es muy clara y que todos los que estamos más o menos metidos en el sector somos conscientes de las implicaciones que puede tener la presencia casi ubicua del software en todos los ámbitos de la vida. Yo opino que cada vez más gente se está sumando a la tendencia de dar más importancia a las cuestiones éticas”, concluye Moreno León, quien juzga que “no es descabellado” plantear que los programadores tengan la obligación que comprometerse con un juramento hipocrático del mismo modo que los médicos.