Recibido: 05.03.2019 | **Aceptado:** 15.04.2019

Palabras clave: Algoritmo, codificación, computación, error y programación.

Mil maneras de morir si no sabes programar

HÉCTOR GERARDO PÉREZ GONZÁLEZ
hectorgerardo@uaslp.mx
FACULTAD DE INGENIERÍA, UASLP

Programar una computadora es el proceso de tomar un algoritmo y codificarlo en un lenguaje de programación, pero, ¿qué es un algoritmo? En un contexto amplio, no es más que una secuencia de pasos a seguir para lograr un objetivo.

De esta manera, si nuestro objetivo es cocinar regañadas (galletas típicas potosinas), el algoritmo en cuestión deberá iniciar con la obtención de los ingredientes (harina, huevos, entre otros), las herramientas y el equipo (rodillo, batidora, horno, etcétera). A continuación se describirá la combinación de los ingredientes (siguiendo un orden preciso) y se concluirá con el horneado y para después espolvorear las galletas con una combinación de azúcar y canela. La correcta preparación de este alimento es representada por dicho algoritmo, que si no es seguido correctamente puede derivar en dificultades como que la galleta quede cruda y al ser ingerida produzca un malestar.

Programar es un poco más complicado que cocinar regañadas, pero ambos procesos son —en esencia— lo mismo: Una serie de pasos para lograr un objetivo. Una receta de cocina contiene instrucciones escritas en algún idioma o lenguaje natural (español, inglés, francés, por mencionar algunos). Un programa de computadora contiene instrucciones escritas en algún lenguaje de programación (Java, C, C++, C#, Python, Ruby, entre otros). El resultado de seguir estos dos tipos de algoritmos de forma equivocada puede traer graves consecuencias, incluso la muerte.

Actualmente, la humanidad depende en gran medida de los software de computadoras, pero programarlas no solamente es darles instrucciones a éstas o a los teléfonos inteligentes. Hoy en día se programan las alarmas de los relojes y de seguridad contra robos, los televisores o los controles remotos, y las personas que carecen de los conocimientos o habilidades para lograrlo pueden estar en serias desventajas respecto a quienes sí pueden hacerlo.

Derivado de lo aquí expuesto, surge la siguiente pregunta: ¿acaso todos deberíamos saber programar? Abordaremos las posibles respuestas una vez que analicemos la siguiente premisa: Si no sabes programar, morirás en consecuencia, y para ello hay muchas maneras de morir. La muerte no sólo es biológica, puede morir tu prestigio, tu salud financiera, tus esperanzas de lograr un objetivo y, en el caso más extremo y no poco común, puedes perder la vida u ocasionar la muerte de muchos si tu programa falla.

Para ilustrar este punto, analizaremos algunas maneras en las que una falla de este tipo derivó en una de las variedades de muerte descritas.

Manera de morir #0001
NASA Mariner (1962)

Remontémonos a los inicios de la carrera espacial. Estamos en julio de 1962, ya pasaron cinco años de que la Unión Soviética colocó en órbita al Sputnik I, primer satélite artificial de la Tierra. Un año después, la Administración Nacional de la Aeronáutica y del Espacio (NASA, por sus siglas en inglés: National Aeronautics and Space Administration) fue creada por Estados Unidos de América (EUA). Los soviéticos seguían a la vanguardia, ya que en 1961 colocaron al primer ser humano en órbita: Yuri Alekseyevich Gagarin, a bordo de la nave Vostok 1.

Para los estadounidenses era urgente obtener una victoria en esta carrera. Tras años de investigación, cálculos, construcción y un presupuesto de 80 millones de dólares, la nave espacial Mariner debía realizar el primer viaje interplanetario. Su destino planeado era Venus, su destino final fue el norte del océano Atlántico. Menos de tres minutos después de su lanzamiento, explotó debido a una serie de problemas de direccionamiento derivados de un error de programación. La NASA

Nave espacial Mariner, manera de morir #0001

Computadora Therac 25, manera de morir #0010

reportó que la falta de un guión en una línea del programa de lanzamiento causó el desastre. Por fortuna, no se perdieron vidas humanas, pero la esperanza de muchos de ganar la carrera espacial murió por meses en EUA. Un error tan simple como la falta de un guión en un programa es muy fácil de detectar en la actualidad, pero hace medio siglo los algoritmos se codificaban en tarjetas perforadas y a la ingeniería de software aún le faltaba un lustro para nacer.

Manera de morir #0010
Rayos X (1982)

Ahora estamos al inicio de la década de 1980. La primera computadora personal (PC, por sus siglas en inglés) de la empresa IBM acababa de ser inventada, pero ya aparecían en la escena comercial algunos otros equipos mas grandes como los PDP de Digital Equipment Corporation (DEC, por sus siglas en inglés) o la HP-3000 de Hewlett-Packard.

Algunos hospitales del mundo ya contaban con equipos de radioterapia controlados por computadora, era el caso de Therac, una máquina de rayos X que había salido al mercado unos años antes. Los modelos Therac 6 y Therac 20 no habían presentado problemas, utilizaban una computadora PDP-11 programada en ensamblador, un lenguaje con ventajas de eficiencia pero muy críptico (es decir, que sólo entendían unos pocos).

El advenimiento del modelo Therac 25 tendría consecuencias catastróficas. El diseño físico de la máquina (hardware) no contempló medidas de seguridad, confiaron esta responsabilidad a los programas (software). Los nuevos programadores de la empresa tomaron el código de las máquinas anteriores y lo adaptaron a la nueva, sin agregar las medidas preventivas necesarias.

Durante esa década murieron al menos 12 pacientes y más de 100 sufrieron graves consecuencias, todo ello derivado de sobredosis aplicadas por error de la computadora de control de la Therac 25.

Tras investigaciones se descubrió que el sistema inicial fue desarrollado por un solo programador no experimentado, presentaba severas fallas de seguridad y casi nula documentación.

En esa época aún no existían los ingenieros de software, dado que la primera de estas carreras del mundo se abrió en 1987, en el Imperial College London de Inglaterra; la primera en EUA fue abierta en 1996 en el Rochester Institute of Technology en Nueva York y la primera en México fue creada en 2004 en la Universidad Autónoma de Yucatán.

Por lo anterior, era muy común que a científicos de otras áreas como la física o las matemáticas se les asignara frecuentemente la altísima responsabilidad de desarrollar software.

Manera de morir #0101
Y2K (2000)

A dos décadas de la aparición de las primeras computadoras personales, llegó el cambio de siglo. Habían quedado atrás las heroicas gestas por lograr que exitosos programas se ejecutaran en memorias de tan sólo 64 kilobytes (aproximadamente 64 000 caracteres). El fin de siglo era entonces testigo del uso de computadoras con varios gigabytes (miles de millones de caracteres). Por esa muy baja capacidad de memoria en los sistemas de cómputo primitivos, se había decidido utilizar el formato

mm/dd/aa (dd/mm/aa es en español) para almacenar una fecha, es decir, (al igual que para el día y para el mes) utilizar tan sólo dos dígitos para representar el año.

Al aproximarse el cambio de siglo fue patente la preocupación, ya que éste aparentemente insignificante problema podría tener trágicas consecuencias. Para entenderlo mejor, planteemos el siguiente ejemplo.

Suponga que usted es cliente de un videoclub (empresa del siglo pasado dedicada a rentar películas) y tiene que regresar cierta película el 20 de diciembre de 1999 (20/12/99). A partir de esta fecha, y si el sistema no recibía la película, cargaba a su tarjeta de débito 100 pesos diariamente por concepto de multa. De esta forma, si la película era entregada el 26 de diciembre, el sistema haría la siguiente operación: 26/12/99 menos 20/12/99.

Un buen algoritmo cargaría un total de 600 pesos (6 días x 100 pesos) a su tarjeta. ¿Pero qué pasaba si usted entregaba la película el 20 de enero de 2000 (20/01/00)? Lo primero que haría el algoritmo sería restar el año de entrega (2000) menos el año en que se debió entregar (1999). Debido a que el año se representaba solamente por sus dos últimos dígitos la resta sería: 00-99, lo que resulta en -99. Esto significa que usted habría entregado la película 99 años antes de la fecha límite. En tal caso, el sistema le haría a usted un retiro negativo (es decir, un depósito) de (-99 años X 365 días X 100 pesos) más de tres y medio millones de pesos. No es de extrañarse que los videoclubes hallan muerto.

Problema Y2K (2000), manera de morir #0101

El Departamento de Comercio de EUA estima que su gobierno tuvo que gastar más de 100 mil millones de dólares para corregir todos sus sistemas antes de que llegara el año 2000, o Y2K.

Manera de morir # 1010
Ransomware (1989 a la fecha)

Un código malicioso no es otra cosa que un programa cuyo algoritmo tiene como objetivo intervenir sistemas ajenos buscando un beneficio ilícito. Virus, gusanos y troyanos son algunos de los tipos de programas dañinos a los que nos enfrentamos. Un virus típico se introduce a un sistema como efecto colateral de haber obtenido

Virus Ransomware, manera de morir #1010

archivos (música, videos, etcétera) de manera ilegal por internet. El daño derivado va desde un simple mensaje anunciando el contagio hasta el robo o secuestro de información y la suplantación de identidad.

Un tipo especial de software malicioso es el conocido como Ransomware. La información de la computadora que ha sido víctima de este programa es encriptada (convertida a un código ilegible), pero no robada; para desencriptar esta información del disco duro de la víctima, el secuestrador pide dinero como rescate (ransom). El mensaje de los secuestradores suele ser algo como esto:

> Tus documentos, fotos, videos y bases de datos ya no son accesibles porque han sido encriptados. Nadie puede recuperarlos sin nuestro servicio de desencriptación, puedes desencriptar sólo algunos de tus archivos, pero si deseas recuperarlos todos necesitarás pagar. La cuota es de medio bitcoin dentro de los próximos tres días, si no pagas en ese plazo, el rescate se duplicará y en siete días será imposible recuperar tu información si no recibimos tu pago (Uninstall.org., Mar.15, 2019).

Pareciera ciencia ficción, pero no lo es. Tan sólo en 2018 se pagaron rescates en bitcoins por el equivalente a más de siete mil millones de dólares. Bitcoin es la moneda virtual mas conocida en el mundo, tan sólo en marzo de 2019 un bitcoin valía mas de 70 mil pesos mexicanos. Lo anterior debido al ataque del ransonware conocido como Wannacry, que tuvo más de 200 mil víctimas en más de 150 países.

> Un software contiene instrucciones escritas en algún lenguaje de computación

No hay nada que podamos hacer para evitar este tipo de ataques, ya que el invasor no requiere que la víctima acceda a algún sitio de internet. Todos somos víctimas potenciales de Ransomware y no podemos hacer nada para prevenirlo, salvo aprender a reducir las posibles vulnerabilidades de nuestros equipos de cómputo. La única forma de combatir los efectos del secuestro de información (cuando se presenta) es permitirlo, la protección consiste en contar con copias o respaldos de ésta realizados periódicamente (diariamente, de ser posible).

Programar el respaldo diario de la información completa almacenada en

una computadora requiere de mínimas habilidades de programación. Programar, por ejemplo, una presentación de Powerpoint con ligas entre páginas, una macro de Excel o un conjunto de cartas modelo de Word, son tareas fáciles que cualquier persona podría aprender, lo que nos lleva a nuestra pregunta inicial.

¿Todos deberíamos saber programar?

En años recientes se ha desarrollado la idea de que todas las personas, especialmente los niños, deberían aprender a programar. Seymour Papert, pionero de la inteligencia artificial, de la enseñanza de las matemáticas y creador del lenguaje de programación Logo, solía decir en sus clases en el Massachusetts Institute of Technology (MIT) "Children should be programming the computer rather than being programmed by it", lo que se traduciría como "Los niños deberían estar programando computadoras en lugar de estar siendo programados por ellas" (Papert, 1980).

Esto no es nuevo, ya en 1995 Steve Jobs, fundador de Apple, declaró en una entrevista: "Todos en este país deberían aprender a programar, porque eso te enseña a pensar" (*Entrepreneur India*, 2017).

Recientemente, en enero de 2016 (*Forbes*), el entonces presidente de Estados Unidos de América, Barak Obama, presentó la iniciativa Ciencias de la Computación para Todos, que enmarcó su proyecto de incentivar con cuatro mil millones de dólares el aprendizaje de las ciencias de la computación, en particular de la programación, en toda la niñez de su país.

Conocer las opiniones de estos grandes pensadores de la historia nos debe hacer reflexionar sobre las iniciativas que realmente deberían ser apoyadas para hacer de nuestros países campos fértiles para el progreso de la ciencia y la ingeniería.

Conceptualizar un problema en nuestra mente, idear un algoritmo para resolverlo, programar su solución y, finalmente, ver el resultado, produce una satisfacción y un orgullo incomparable. Si además de esto, facilita la vida e incluso nos salva de perderla, valdría la pena iniciar este camino, y así evitar ser víctima de alguna de las mil maneras de morir si no sabes programar. UP

Referencias bibliográficas:

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Nueva York: Basic Books, Inc.

Hooda, S. (Febrero 16, 2017). #3 Reasons Why Everyone Should Learn Programming. *Entrepreneur India*. Recuperado de: https://www.entrepreneur.com/article/289248

Shapiro, J. (31 de enero, 2016). President Obama Wants Every Kid To Learn Coding—For All The Wrong Reasons. *Forbes*. Recuperado de: https://www.forbes.com/sites/jordanshapiro/2016/01/31/president-obama-wants-every-kid-to-learn-coding-for-all-the-wrong-reasons/#4921cdb969af

Douafi, R. (16 de julio, 2018). Hear to remove Blackfireeye ransomwere and decrypt jes files. Uninstall.org (Mar 15, 2019)

HÉCTOR GERARDO PÉREZ GONZÁLEZ

Doctor en ciencias de la computación por la Universidad de Colorado. Actualmente es profesor investigador en la Facultad de Ingeniería de la UASLP y lidera el proyecto "Museo Interactivo de Tecnología Aplicada (MITA)", apoyado por el Conacyt.