Desafíos Jurídicos y éticos de los algoritmos desde su creación hasta la destrucción

Iván Díaz González

Durante varias décadas hemos visto la evolución de los medios electrónicos involucrándose fuertemente en la vida cotidiana de la humanidad, abriendo espacio para la integración de las computadoras en diversas actividades, sin embargo la posibilidad de esta interacción entre las computadoras y las actividades humanas no sería posible sin la definición de un conjunto de pasos que permitan reconocer la necesidad del ser humano en un entorno particular para resolver un problema específico, de ahí que se requiera de un algoritmo para la actuación del componente electrónico para su interacción con el ser humano.

Un algoritmo por definición es "*Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema*" (Real Academia Española, s.f.), de esta definición podemos resaltar las palabras "Conjunto ordenado y finito de operaciones" ya que ha sido muy común establecer que para llegar a un mismo resultado y que este sea comprobable se deben establecer estas características; tal es el caso, del método científico.

Los algoritmos están presentes en muchas de nuestras actividades cotidianas, desde la receta de cocina, el manual de instalación de una televisión o más sofisticados como las macros de una hoja de cálculo o bien pueden estar integrados en programa de cómputo que permiten que se ejecuten ciertos pasos de forma automatizada por los componentes electrónicos o por las computadoras, sin

embargo no contemplan todas las características de las problemáticas sociales.

Existen problemas de la vida cotidiana, tales como la comunicación entre los individuos, en donde sabemos que existe un emisor, un receptor, un mensaje y un canal o medio de comunicación, en el caso de la comunicación se han dado soluciones de forma evolutiva modificando el medio de comunicación para eficientar el proceso de comunicación en grandes distancias en menos tiempo dando lugar al correo, telégrafo, teléfono, correos electrónicos, chats, entre otros, estableciendo con ello, nuevos paradigmas que impactan las necesidades sociales.

Los cambios en los medios de comunicación resuelven un problema con el canal dejando de lado otros componentes básicos establecidos en la teoría de la comunicación como el emisor y el receptor que comúnmente se relacionan por concepto a una entidad humana, provocando así una construcción social que limita el entendimiento del involucramiento de un componente electrónico en este proceso.

La integración de los componentes electrónicos en actividades repetitivas que pueden suplir la intervención humana, si bien pueden beneficiar a ciertos sectores sociales al permitir reducir los costos a través de la automatización, por otro lado pueden poner en riesgo a ciertos miembros de la sociedad que no conceptualizan las amenazas que prevé el entorno, en el ejemplo de la comunicación riesgos como la identificación de la otra persona, la alteración del mensaje, la perdida de la comunicación o algunas otras, son algunos ejemplos de riesgos que provocan ataques más conocidos como suplantación de identidad, phishing, robo de sesión, etcétera.

En el ámbito tecnológico los algoritmos han tenido una gran aportación para la creación de los programas de cómputo que han sido implementados durante años para la administración de las empresas en las áreas de finanzas, contabilidad, recursos humanos, ventas, producción, variando las implementaciones según los sectores de mercado y con ello mejorando sustancialmente los tiempos de ejecución de actividades que naturalmente son repetitivas y con entradas y salidas de datos que se encuentran ampliamente estandarizados y controlados.

Pero la integración de esto programas de cómputo no es estática, desde hace algunos años ya no solo interactúan con procesos de entradas y salidas definidas, sino que se ha llevado a un nuevo nivel en donde se intentan procesar textos escritos a mano, permitir la atención a clientes por vía de una comunicación más común a través de herramientas que contemplan toma de decisiones automatizadas sin intervención humana valorativa y que a la

par puede modificar sustancialmente el conjunto de entradas y salidas de datos, sin que con ello se modifique el proceso en su generalidad, desde la perspectiva en que se segmentan las actividades en procesos mínimos que permiten una interacción libre de cambios y basados en el algoritmo principal.

La complejidad de los algoritmos humanos requiere de una microsegmentación de las actividades con objetivo particular para que con ello se pueda establecer el conjunto de pasos, así como las entradas y salidas de información, ya que sin esta microsegmentación no sería tan factible la interacción, ejemplo de ello es el proceso de atención a clientes de una empresa con un chatbot con inteligencia artificial, de primera instancia es importante declarar el objetivo del proceso que en este caso es atender la llamada, para lo cual el programa de cómputo cuenta con actividades definidas para abrir el canal de comunicación y con algún método introducir la conversación, tal y como lo haría un ser humano.

El siguiente paso es reconocer cual es la necesidad del cliente, en este punto el algoritmo solo debe ir encaminado a reconocer la necesidad mediante un conjunto de análisis de palabras o de instrucciones que el cliente comunica durante la conversación, en el caso de del ser humano la interpretación de las palabras y el tono de la voz ayudaran en la toma de decisión de cómo ayudar al cliente, sin embargo cuando lo realiza un programa de cómputo las entradas diversas tienen un conjunto de posibilidades que llevaran al algoritmo a generar una salida con respecto al conjunto de información recolectada durante la conversación, en cualquiera de los dos casos con o sin intervención humana la actividad del algoritmo es clara identificar la necesidad del cliente.

Con este ejemplo podemos evaluar que, con una identificación plena del objetivo, el algoritmo no se modifica en su estructura lo que permite establecer las actividades y homogenizar los resultados, lo que puede cambiar es el conjunto de entradas y salidas con base en la toma de decisiones y los factores que probabilidad que se tienen con respecto a la percepción del cumplimiento del objetivo.

Ahora desde la perspectiva jurídica los algoritmos pueden apoyar en gran medida en la estandarización u homogenización de los resultados con respecto al objetivo, sin embargo, existen ciertas restricciones en claridad en la que el proceso se está llevando a cabo por lo que se requiere de certidumbre en la trazabilidad de cada una de las actividades del algoritmo, así como claridad sobre las entradas de los flujos de información que se tuvieron para que el algoritmo procesara la información.

En el contexto físico con intervención humana un ejemplo de ello es la firma de un contrato, un algoritmo ampliamente aceptado es la firma de un contrato bajo la fe pública, en donde un tercero de las partes firmantes hará un conjunto de actividades para identificar a las partes plenamente, en su caso dar lectura al contrato y por ultimo recabar las firmas cerciorando los trazos a su percepción, ahora si este algoritmo lo queremos llevar al ámbito tecnológico el proceso seguirá siendo el mismo, solo que ahora la validación no estará a cargo de una persona sino de un programa de cómputo que requiere de una ingesta de información significativa que le permita identificar a los individuos tal como identificar los rasgos faciales, también puede dar lectura al contrato y recabar las firmas, lo cual no está alejado de las actividades establecidas en el entorno físico.

Si bien, el objetivo planteado en el algoritmo se puede cumplir plenamente los programas de cómputo entre muchas de sus características se encuentra una distribución en múltiples capas que impactan tanto a la información de entrada y salida del propio algoritmo, pero sin la debida regulación puede impactar otros algoritmos que no están previamente resguardados, ejemplo de ello es que en un proceso de compra en línea el algoritmo para la venta recolecta información personal para la compra, recolecta la información del producto, recolecta información del método de pago y por ultimo procesa la compra, este proceso cumple con el objetivo de la venta, sin embargo pueden existir conectados otros algoritmos que guarden la información del método de pago y que a su vez lo transfieran a un tercero o bien que analicen información que está contenida en el equipo de cómputo desde el cual se hizo la compra (cookies).

Otro caso que se puede presentar en el ámbito jurídico es la discriminación, ya sea por errores propios del algoritmo o por la información de entrada para la toma de decisiones, sin importar el motivo del error, la afectación del algoritmo tendrá un impacto en la esfera jurídica de la o las personas que interactúan a través del algoritmo, ejemplo de ello es el otorgamiento de las líneas de crédito, que sin importar si se hace o no con intervención humana, dependerá de la información que se tiene al momento, así como el proceso de validación del historial crediticio, lo cual sin lugar a duda afecta al solicitante y al evaluador ya que el primero no obtendrá el beneficio del crédito y el otro perderá un cliente en el peor de los escenarios.

También en el ámbito jurídico es necesario atender la autonomía humana, ya que los algoritmos intentan homogenizar el procesamiento y la salidas con respecto a un conjunto de entradas, esto impide que la

sensibilidad humana se elimine como parte de las variables del proceso, lo cual en algunos procesos puede ser positivo ya que impide el sesgo y por consiguiente la posibilidad de errores en el proceso, sin embargo existen condiciones que no están claramente identificadas que al no ser parte del algoritmo no serán evaluadas y por consiguiente no formaran parte de la toma de decisiones cuando se encuentre regulada por un algoritmo, lo que conlleva una decisión sesgada por si misma y que a través de la autonomía humana permitirá la integración de una forma simple.

Ahora bien desde la perspectiva ética es imprescindible analizar el procesamiento de los algoritmos para la intervención en la toma de decisiones sin intervención humana valorativa, en donde no habrá espacio para el análisis del contexto situacional del proceso o de los elementos que no son parte del análisis del algoritmo o que no fueron tomadas en consideración en el momento de su concepción y diseño lo que conlleva a que la necesidad de establecer un conjunto de normas éticas que establezcan un punto de partida referencial para la integración de los modelos de trabajo, de ahí podemos analizar como marco de referencia los principios de Asilomar (Robotechnics, 2017) que determinan un conjunto de características aplicables a la inteligencia artificial, aunque si es necesario entender que este trabajo no solo habla de la regulación algorítmica en el uso de inteligencia artificial o máquinas de aprendizaje sino una visión holística de los usos de este concepto.

Los enfoques que se han descrito hasta el momento están referenciados a un estudio de la Unión Europea para la confiabilidad de la Inteligencia Artificial (Comisión Europea, 2018), ya que ayuda a identificar que si bien el uso de los algoritmos a través de las tecnologías de la información y los programas de cómputo no detendrán su inercia de interacción con la sociedad y los individuos, si generan una brecha de confiabilidad en la integración y el uso en situaciones complejas tales como la impartición de justicia, la administración de los recursos y otros aspectos que tienen un impacto en las creencias de la humanidad.

1. **¿Qué es un algoritmo?**

Como ya se ha elaborado anteriormente, la Real Academia de la Lengua Española lo define como "*Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema*" (Real

Academia Española, s.f.), también podemos encontrar la definición de NIST que dice que es "*Un conjunto computable de pasos para lograr el resultado deseado.*" (Black, algorithm, 2020), del diccionario de Cambridge se puede obtener la definición "*un conjunto de instrucciones o reglas matemáticas que, especialmente si se dan a una computadora, ayudarán a calcular la respuesta a un problema*" (Cambridge, s.f.).

Analizando las definiciones que se encuentran en diversas fuente podemos encontrar que un punto de referencia de los algoritmos es que existen un conjunto de pasos, este conjunto de pasos podría ser desde cómo preparar una receta de cocina o bien como prepararnos una taza de café, sin embargo se puede complejizar la estructura de la identificación de los pasos cuando tenemos una condición humana intermedia, en el caso de la receta de cocina las variantes podrán encontrarse en el gusto de las personas por un ingrediente especifico y su forma de integración, pensemos en la pimienta habrá quien le guste la pimienta previamente molida y la que prefiere que la pimienta este fresca y molerla por cuenta propia, si bien no cambia la estructura de los pasos para la receta por si mismo, si cambia sustancialmente la forma de preparación de la receta.

También es importante establecer en este orden de ideas que los pasos que se llevan a cabo también pueden variar en el contexto ambiental en el que se están desarrollando, retomando el ejemplo de la receta no será lo mismo preparar la receta en una cocina con elementos electrónicos y con entera disposición que prepararlo en un fogón o en condiciones rurales en las cuales se pueden excluir herramientas que pueden simplificar el conjunto de pasos.

El conjunto de pasos que se requieren atender también estarán supeditados a las creencias del ejecutante y las condiciones sociales en las que se desenvuelvan por lo que para agotar el ejemplo de la receta pensemos en una receta que está en condición para que la carne se sirva a término medio, pero por la condición de las creencias del ejecutante requiere que el término de la carne sea bien cocido, lo cual no cambia el objetivo pero si modificara el conjunto de pasos y las variables del entorno en la cual se están ejecutando.

Los pasos que se integran a un algoritmo deben permitir establecer el conjunto de características que permitan una homogenización de las operaciones para permitir que se vuelvan repetibles a través de cada una de las ejecuciones con el objetivo similar o igual con lo cual se permite establecer la definición de un proceso libre de exclusiones y solo se denotara manejo de las variables de entrada para la ejecución optima de los pasos.

Si lo queremos llevar a un proceso computacional, es común ver requerimientos específicos de los usuarios cubriendo necesidades particulares para un objetivo que es general pero con variables de entorno muy específica, lo cual tiene como resultado que el algoritmo se encuentre alineado exclusivamente para las condiciones establecidas y excluyendo con ello cualquier otro elemento que no esté contemplado como caso de uso, para poner un ejemplo pensemos en un algoritmo para crear un contrato de arrendamiento simple en la Ciudad de México entre dos personas físicas, con estas condiciones el algoritmo de creación del contrato estará supeditado a integrar la información de las personas físicas, el bien inmueble que se arrendará y el marco regulatorio correspondiente a la Ciudad de México, sin embargo este elemento puede tener una modificación en su conceptualización y establecer que se creen contratos de arrendamiento entre dos partes en cualquier lugar de la República Mexicana lo cual modifica el algoritmo, ya que se requiere identificar si alguna de las partes es persona física o moral, se requiere establecer el estado de la República y toda la información necesaria para poder crear el contrato.

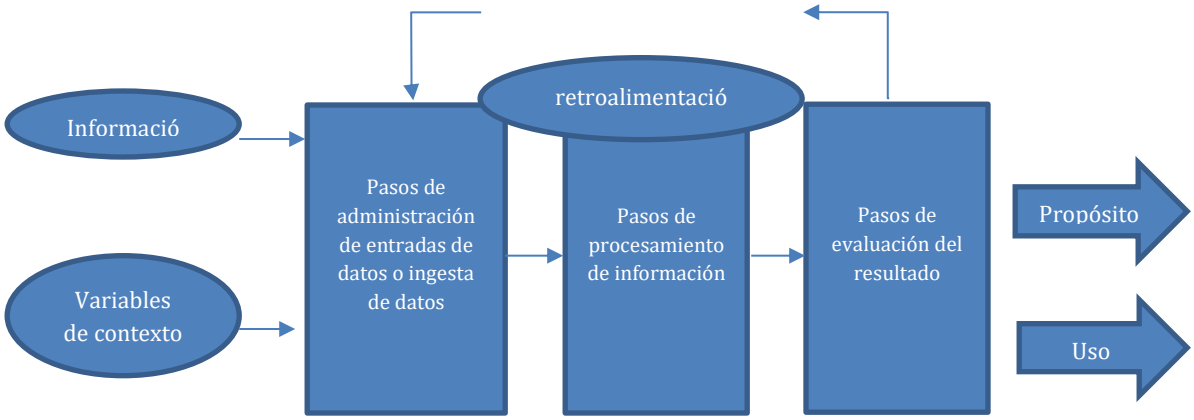
Con el ejemplo anterior notamos que la definición del objetivo o del resultado requerido es un elemento importante en la definición de los pasos, es por ello por lo que de todas las definiciones de algoritmo se desprenden las palabras resultado de un problema, objetivo específico o cualquier otra relacionada, ya que la definición de este elemento ayudará a la interpretación y definición del conjunto de pasos para solucionar el planteamiento.

Un elemento imprescindible dentro de los algoritmos es la definición del cumplimiento del objetivo o la resolución del problema, ya que si no se cuenta con un mecanismo que pueda identificar que se haya cumplido con el objetivo entonces el algoritmo se quedará naturalmente inconcluso al no poder establecer el término del conjunto de pasos para lo cual se requiere de una retroalimentación que evalúe el cumplimiento del objetivo.

Pero el planteamiento del problema o el objetivo no lo es todo, ya que una vez que un conjunto de pasos que cubren con un objetivo en particular, pueden ser alineados a través de modificaciones en las variables de entrada o del contexto para resolver otras problemáticas con ciertas similitudes, lo cual en la perspectiva general puede tener un impacto positivo, ya que permite reutilizar los esfuerzos de un algoritmo preestablecido en múltiples objetivos, sin embargo también existe la posibilidad que el conjunto de pasos realizados con un objetivo negativo se vuelva más sencillo de ejecutar, ejemplo de ello es el SPAM, de primera

instancia se han desarrollado algoritmos para hacer la entrega masiva de correos electrónicos que permiten comunicar de forma más rápida y eficiente la información en un modelo de difusión simple, el propósito positivo es cambiar el objetivo y mandar publicidad a todos los clientes de una empresa, pero que pasa cuando un empleado o un agente malicioso utiliza el servicio para enviar malware o correos que desprestigian a la empresa, no cambia por sí mismo el algoritmo ni el objetivo, lo que cambio es el propósito y el uso que se le dio al algoritmo, lo cual se puede volver un elemento que no estaba claramente identificado.

Por todo lo anterior, para poder definir a un algoritmo desde una perspectiva técnica y jurídica no solo puede ser un conjunto de pasos con un objetivo particular ya que tiene una variedad de entradas de información, entradas del contexto, esquemas de retroalimentación que se vuelven parte del conjunto de pasos y también se deben de agregar al objetivo el conjunto de usos y propósitos para los que ha sido establecido, diseñado y creado el algoritmo, a continuación se presenta un diagrama genérico de la conceptualización de un algoritmo.



Con este diagrama se estructuran los componentes de un algoritmo de forma general y que una vez que se definen en el espacio del entendimiento humano también se pueden llevar a un modelo de ejecución mediante componentes computacionales que pueden hacer más eficiente la ejecución reduciendo los tiempos de respuesta y limitando los casos de uso respecto al conjunto de entradas que están a disposición en el análisis de la etapa inicial de ingesta.

Con base en el conjunto de elementos estudiados del algoritmo es necesario

determinar los tipos de algoritmos para especificar los rasgos característicos que delimiten el rumbo normativo en su aplicación.

Tipos de algoritmos y sus requerimientos de normalización.

De primera instancia podemos definir que tenemos algoritmos determinísticos y no determinísticos que suena a una definición muy técnica pero que permite organizar el espacio de análisis de los algoritmos dependiendo el comportamiento definido de los pasos ejecutados con respecto al resultado, dicho en palabras más simples los algoritmos determinísticos son aquellos en el cual el comportamiento es predecible conociendo las variables de entrada (Black, deterministic algorithm, 2009), ejemplo de este tipo de algoritmos determinísticos es cuando seguimos un conjunto de pasos para guardar un archivo en la computadora, si seleccionamos la carpeta, ponemos el nombre y por ultimo oprimimos el botón guardar, sabemos que no importa la cantidad de veces que se ejecute el algoritmo siempre el resultado será que se almacenara la información en el disco, con un nombre que es parte de las variables y en una ubicación que esta determinada como parte del proceso mediante una variable.

Por otro lado, tenemos los algoritmos no determinísticos son aquellos que permiten más de un solo paso en determinados momentos y que siempre busca ejecutar el paso correcto o el mejor (Black, nondeterministic algorithm, 2020), lo cual permite que a pesar de que se tengan las mismas entradas se puede generar múltiples salidas diferentes dependiendo el momento de la ejecución. Un ejemplo de un algoritmo no determinístico es la atención de un cliente por medio de un chatbot en donde se abren un conjunto de posibilidades en cada una de las interacciones del cliente con el cliente y a pesar de que se sigan los mismos pasos, no en todos los casos el resultado será un elemento definido, aunque será el mejor con respecto a las interacciones.

Otro tipo de algoritmos son los aleatorios que incluyen como parte de su estructura variables o componentes dentro de la ejecución que son generados de forma aleatoria o pseudoaleatoria (Black, randomized algorithm, 2019) y su uso es comúnmente para llevar a cabo simulaciones dentro de un proceso como semillas para hacer la comprobación de multiplicidad de eventos o bien estos se pueden utilizar en gran medida en procesos para apuestas en donde la aleatoriedad juega un papel significativo.

Existen otros tipos de algoritmos que pueden representar algún cambio en el modo de ejecución o de entradas y salidas sin embargo no es la finalidad de este documento abundar en los tipos. Con la definición de los tipos, es más sencillo encontrar un punto de

homologación en el entendimiento de las necesidades en la normalización de los algoritmos, que será un punto de partida para la especificación de un conjunto de parámetros o métricas que pueden ser evaluadas de forma general.

Algoritmos determinísticos

Para el caso de los algoritmos determinísticos puede ser sencillo establecer las métricas, ya que se espera un conjunto de ejecuciones específicas y que no cambian con respecto al tiempo ni a los modos de ejecución, en este sentido uno de los primero elementos a tratar es la calidad sobre el procesamiento de la información y para ello se requiere controlar la entrada de los datos y el despliegue de los componentes de programación para el procesamiento de información, ya sea desde una perspectiva de seguridad de la información, protección de datos personales o meramente procedimental.

Para el caso de la entrada de datos no es nuevo que en términos de seguridad sea uno de los puntos más significativos en los procesos de aseguramiento del software, de ahí que existan como mecanismos de evaluación en diferentes marcos de referencia el manejo de las entradas de datos.

El manejo de las entradas de datos está basado en hacer una inspección de que la información que se procesara en cada una de las etapas del sistema de información no contenga información que pueda provocar una alteración al procesamiento natural de la información, tal y como pasa en los ataques de inyección de SQL, XML Entities o XSS(Cross Site Scripting) (The Open Web Application Security Project, 2010), en los cuales una entrada de datos al sistema de información puede provocar que se altere la operación del sistema. (The Open Web Application Security Project, 2017).

Para ésta validación que se está tomando en consideración a través de las propuesta de revisión de los marcos de referencia en materia de seguridad se requiere de una validación de la sintaxis con la cual se integra el modelo de procesamiento de información, el ejemplo más claro de ello es que no se deben permitir las entradas de correo electrónico que no lleven palabras, el símbolo “@” y después una palabra con un punto y una siguiente palabra tal como prueba@correo.com, ya que sintácticamente cualquier cosa diferente no será un correo electrónico.

Pero también hay otras revisiones que se deben realizar y es que para evitar que se altere el flujo de la aplicación no puede haber intervención de palabras clave de lenguajes comunes de programación en SQL o modificaciones de etiquetas que permitan agregar comandos a la ejecución, para lo cual se

deben utilizar mecanismos de entrada de datos que no modifiquen su funcionamiento a pesar de que las entradas tengan estos elementos incluidos o bien eliminar estas palabras y símbolos en alguna parte del proceso.

Esta entrada de datos también nos puede llevar a establecer necesidades de operación en materia de protección de datos personales, ya que en el reglamento de la Ley Federal de Protección de Datos Personales en Posesión de Particulares se establece el principio de Calidad que hace referencia a "los datos personales tratados sean exactos, completos, pertinentes, correctos y actualizados según se requiera para el cumplimiento de la finalidad para la cual son tratados" (Cámara de diputados del H. Congreso de la Unión, 2011), por lo cual el asegurarnos que la entrada de datos cumpla con estas características se convierte en una obligación en cuanto a tratamiento de datos personales se refiere.

Para cumplir con los preceptos establecidos en términos de la calidad, no solo basta con la validación de la información en forma sintáctica, sino que también se requiere evaluar la semántica de la información con respecto al contexto bajo el cual se recaba la información, ya que en caso contrario el algoritmo puede procesar información de forma incorrecta y con ello incumplir con el precepto de calidad.

Ahora en el caso del procedimiento se requiere que de primera instancia existan un conjunto de reglas de operación claramente establecidas, codificadas y posteriormente probadas, pero también es indispensable asegurar que de la misma forma en la que se probó el proceso que se desarrolló en algún momento, también cumple con las mismas características del despliegue y uso de la información.

Este proceso esta alineado a la necesidad de cumplir con lo que se denomina gestión de cambios y despliegues, lo que permite que haya integridad de información entre lo que se conceptualiza como necesidad en la creación de un algoritmo programado y lo que se utiliza por lo usuarios finales para el procesamiento de información. Se pueden tener procesos como DevOps (Desarrollo-Operación) o DevSecOps (Desarrollo-Seguridad-Operación) como herramientas para lograr los objetivos que se buscan en este apartado, sin embargo no es propio de este artículo el estudio de estas herramientas, pero si su relación con el ciclo de vida de desarrollo de software en el cual nos dará como resultado un seguimiento apropiado de cada uno de los pasos para la protección de la ejecución de los algoritmos.

Algoritmos no determinísticos

Los algoritmos no determinísticos, no cuentan con una relación especifica de la salida que se obtendrán del procesamiento de la información

a una misma entrada de información, lo que establece como punto de partida una necesidad sobre la trazabilidad y la explicación de como el algoritmo opera y también que elementos tomó en el tiempo de ejecución, por lo que una variable importante en este proceso es el tiempo en el que se toma la decisión.

En este sentido, los algoritmos no determinísticos deben ser evaluados desde una perspectiva de su momento de ejecución, lo cual en el ámbito jurídico establece un elemento de control que conlleva un requerimiento para establecer la condición de la operación y que está basado en el conjunto de datos, probabilidades y contexto en el cual se produjo la acción para la toma de decisiones o la acción misma del algoritmo.

Para ponerlo en un ejemplo, pensemos en un algoritmo no determinístico que tomará la decisión en un proceso penal, para lo cual la entrada de datos estará en consideración de los hechos y las reglas que están establecidas en el entorno de la jurisdicción de la que se trate, las normas aplicables y el contexto en el que se desarrolla la decisión, sin embargo en la primera salida del proceso la decisión puede ser la determinación de la culpabilidad y por ende la punibilidad de la acción, pero si este proceso lo ejecutamos en diversas ocasiones con las misma características con una retroalimentación sobre las decisiones tomadas y con la integración de criterios probabilísticos de la operación de los hechos, se pueden generar modificaciones en cada una de las decisiones, ya que dependerá de la ponderación de las variables.

De lo anterior, surge la necesidad de preservar intactos los derechos de los individuos que se pueden ver impactados por las decisiones de los algoritmos determinísticos, por lo que en la normatividad mexicana se cuenta con la posibilidad de rectificación en el reglamento de la LFPDPPP (Cámara de diputados del H. Congreso de la Unión, 2011), pero la rectificación no sería posible sino se cuenta con la explicación del porque se tomó la decisión, lo cual establece la necesidad de contar con un modelo de entendimiento de la operación del algoritmo y que se exhibe en el artículo de “Uso responsable de la IA para las políticas públicas: Manual de Ciencia de datos” lo exhibe como el principio de explicabilidad (Gonzalez, Ortiz, & Sanchez, s.f.), en el cual se hace preponderante conocer cómo es que opera el algoritmo.

Uno de los mayores usos que se les da a los algoritmos no determinísticos se encuentra alineado a la inteligencia artificial, de ahí que su estudio se confunda por la relación tan estrecha que existe entre estos dos conceptos.

En este orden de ideas también es indispensable tomar en consideración que las ejecuciones de los algoritmos de este tipo

pueden impactar no solo los derechos de los individuos sino su autonomía en la toma de decisiones o en sus acciones por lo que se requiere evaluar el nivel de confianza de los algoritmos para poder implementarlos para la aplicabilidad humana, de ahí que la integración de los elementos de robustez, legalidad y ética establecidos en las directrices de confianza establecidas por la Unión Europea (Comisión Europea, 2018), juegan un papel importante.

Si bien la Unión Europea plantea sus directrices de confiabilidad basado en tres ejes tan importantes, también es posible segregar más estas características de cómo deben operar los algoritmos con inteligencia artificial a través del conjunto de principios planteados en la convención de Asilomar (Robotechnics, 2017), en donde se plasman 23 principios que ofrecen un gran acervo de conocimiento aplicado a esta materia.

Algoritmos aleatorios

En este tipo de algoritmos uno de los elementos más significativos es la definición de la aleatoriedad debido a que se espera que el resultado de la operación no sea predecible, ya que si bien hacer que un evento o una variable sea aleatoria suena como una tarea sencilla, en la realidad solo es una percepción que puede estar causada por el desconocimiento de las operaciones que generan el resultado.

En el ámbito computacional, todas las operaciones que se realizan están basadas en los ciclos de reloj del procesador por lo que es completamente predecible calcular un numero “aleatorio” en una computadora y lo cual resta valor a la calidad de los algoritmos aleatorios y se genera el concepto de pseudoaleatorio.

En el ámbito jurídico la relevancia que puede tener esta diferencia de tener o no un algoritmo complemente aleatorio se encuentra alineado más a la relación de los juegos de azar y las obligaciones que se generan y también en la exposición de estos algoritmos para cálculos de eventos que pueden provocar un riesgo.

En general, la regulación de los algoritmos no solo está supeditado a la forma de procesamiento de la información y se requiere que se evalúe en entorno general de cada uno de los elementos que interactúan para el objetivo sobre el que fue creado y con ello buscar mecanismos para evitar que se modifique el propósito sin control para afectar otros aspectos que no se encuentran definidos de forma principal.

Por otra parte, se deben evaluar las entradas de información, lo que significa que deben existir modelos de gobernanza de datos que acrediten el cumplimiento de la normativa en protección de datos, privacidad, derechos de autor, derechos humanos y patrimoniales que puedan estar relacionados.

Tampoco se puede dejar de lado la evaluación ética en la administración de la información, ya que de ello depende que este conforme a las necesidades sociales y que se alinee al conjunto de creencias, valores y actividades de la sociedad en donde se desempeña el algoritmo.

Pero no se debe dejar de lado que el procesamiento de la información requiere de un proceso de desarrollo del algoritmo que de una u otra forma será el elemento que ejecutará las actividades y que para ello se requiere de administrar la forma en la que se definen las reglas de operación hasta su liberación para el uso de las personas, por lo que en el siguiente apartado se estudiaran algunos aspectos relacionados con el ciclo de vida del desarrollo de software.

3 Ciclo de vida de desarrollo de software

Los algoritmos pueden ser gestionados desde su creación a través del ciclo de vida de desarrollo de software, ya que es una herramienta procedimental que permite dar seguimiento desde el requerimiento hasta la liberación y desecho de un proceso que se automatiza o que se lleva a un sistema de información orientado a un programa de cómputo, desde esa perspectiva su estudio ayuda como mecanismo de control en cada una de las etapas para normar las características de los algoritmos en sus componentes.

El ciclo de vida de desarrollo de software es el proceso mediante el cual se puede llevar a cabo la integración de los componentes tecnológicos, procedimentales y de personas parala entrega de un producto de software que cumpla con los requerimientos de los agentes solicitantes para las funciones previstas durante la su concepción.

En muchos de los casos las operaciones o funciones están previstas desde la perspectiva de operabilidad o usabilidad de estas características y que como parte del proceso de transformación al entorno digital se llevan a cabo una serie de pasos o actividades para cumplir con ese objetivo.

Los pasos para llevar a cabo son muy variados dependiendo la metodología que se utilice ya que dependiendo la necesidad y/o las características del proyecto se puede definir metodologías en cascada, prototipos, incrementales, espiral, sala limpia, Joint Application Development, Rapid Application Developmet o Agile, en este último punto el Agile puede contar con sub-metodologías que pueden cambiar significativamente las características del proceso.

Sin embargo, por el tiempo en el mercado y su amplio uso las metodologías en la cual podemos encontrar la mayor cantidad de pasos que describen el ciclo de vida de software es la de

cascada que determina las siguientes actividades:

Inicial: si bien no es una fase por sí misma del desarrollo, de esta etapa se desprende la necesidad y el caso de negocio para la determinación de la viabilidad

Requerimientos: se recupera y documentan los requerimientos de los usuarios con respecto a las necesidades

Arquitectura: esta actividad puede estar en conjunto con el diseño, sin embargo, su análisis por separado permite evaluar la integración a las estrategias de la organización.

Diseño: en esta etapa los requerimientos se transforman en un elemento más tangible de lo que posteriormente será un software operativo

Construcción: se implementa la arquitectura y diseño en un código para que se aprecien las funcionalidades

Pruebas y evaluaciones: es un mecanismo de control que permite evaluar el cumplimiento de las necesidades y requerimientos funcionales y no funcionales.

Liberación y mantenimiento: se pone en producción el software para que pueda ser utilizado por los usuarios con base en las necesidades planteados, en el mantenimiento se verifican a través de controles los cambios o mejoras que se requieren con base en nuevas necesidades o requerimientos

Eliminación: esta etapa es poco común verla en los procesos de desarrollo de software sin embargo también es parte del ciclo de vida, ya sea por reemplazo o porque la necesidad ya no se encuentra activa, esta etapa debe ser tomada en consideración para contar con controles establecidos durante la vida de un software

Hasta este punto se ha hablado únicamente del ciclo de vida de desarrollo sin tomar en cuenta la seguridad que actualmente juega un papel muy importante en las necesidades de las organizaciones desde diversos puntos de vista por ejemplo en las necesidades de uso de la organización, cumplimientos técnicos y también cumplimientos normativos en el procesamiento de información tal es el caso del GDPR de la Unión Europea, en donde se establece la característica de seguridad desde el diseño que si bien no está completamente dirigida a desarrollo de software, si se alinea a parte de los procesos en la conceptualización de sistemas para el procesamiento de información. (Parlamento Europeo y Consejo de la Unión Europea, 2016)

En este sentido el S-SDLC tiene la finalidad de integrar en cada una de las etapas del proceso del ciclo de vida de desarrollo de software los controles necesarios para el aseguramiento de

la información del producto final de software, para ello cada una de las etapas tendrá elementos de valor significativos en esta definición con lo son:

Requerimientos: obtención de requerimientos de información que comúnmente son determinados como características no funcionales a menos que el software tenga como principal funcionalidad la administración de la seguridad

Arquitectura: permite la identificación de amenazas y los planes de tratamiento en consideración de las actividades de la organización, así como de su estrategia.

Diseño: permite la implementación de los modelos de gestión de las amenazas, también permite la definición e identificación de amenazas con base en las características del modelado

Construcción: en esta etapa se cuentan con diferentes modelos que permiten integrar código que cuente con las características de seguridad al determinar líneas base de elementos que cumplan con las características del diseño y arquitectura planteados

Evaluación y pruebas: se puede realizar pruebas específicas sobre elementos requeridos y arquitectados pero también se pueden hacer pruebas para detectar elementos que no habían sido concebidos como parte del proceso

Liberación y mantenimiento: en esta actividad la seguridad está enfocada a salvaguardar el software de errores por la administración de los servicios en los procesos de despliegue y cambios.

Eliminación: en esta fase la seguridad se enfoca en mantener los datos protegidos durante el proceso y después del mismo, haciendo que los elementos que procesaron o almacenaron la información no puedan ser utilizados para afectar algunos de los principios de la seguridad. (Open Web Application Security Project, s.f.)

Como se puede validar la seguridad es parte integral del ciclo de vida de desarrollo de software siempre que se opte por una estrategia en el sentido de salvaguardar la información ya sea por convicción propia o por normativas regulatorias.

3.1 Descripción resumida de los diferentes tipos de S-SDLC

Mas que tipos son marcos de referencia de ciclos de vida de desarrollo de software seguros y en este sentido es importante describir algunos que son desde mi punto de vista sumamente relevantes por sus características:

OPENSAMM: es un marco de referencia de OWASP que describe un modelo de madurez para la implementación y evaluación de una estrategia de seguridad en el software, sus

características trascienden a la capa del ciclo de vida natural del desarrollo exponiendo una capa de gobierno que interactúa con la estrategia de la organización, lo que permite que sea un marco de referencia con gran profundidad y alineado a los objetivos de la organización y que a su vez sea medible. (Open Web Application Security Project, s.f.)

NIST SP800-64: si bien este marco de referencia está alineado a una estrategia de ciclo de vida de desarrollo de sistemas de información y no únicamente al desarrollo de software cuenta con elementos fundamentales que no se toman en consideración en otros marcos de referencia por su exclusivo uso en materia de desarrollo, algunas prácticas que se pueden encontrar documentadas es la adquisición de software de terceros y su integración y asimismo en los elementos arquitectónicos se puede evaluar una amplitud de posibilidades ya que no está supeditado a únicamente ver el ciclo de vida en específico con la codificación sino también con la integración de múltiples partes como se vive en una realidad más actual.| (National Institute of Standars ans Technology, 2008)

PCI SLC: este marco de referencia está orientado a la protección del software desde una perspectiva financiera y también se monta en las características de un modelo de gobierno y cambia algunas características del ciclo de vida por dominios de control mucho más abiertos y que evalúan las diversas etapas del ciclo de vida desde una perspectiva más funcional (Payment Card Industry, 2021).

ISO 27034: este estándar de la serie de ISO 27000 es un documento que especifica un modelo de ciclo de vida de desarrollo seguro basado en procesos y guía en la forma de contar con aplicaciones seguras.

ISO 12207: este estándar establece un proceso de ciclo de vida del software y está dividido en tres grupos de procesos que se describen a continuación:

Primario: adquisición, suministro, operación, mantenimiento y destrucción

Soporte: auditoria, administración de la configuración, documentación, aseguramiento de la calidad, solución de problemas, verificación y validación

Organizacional: Administración, gestión de infraestructura, mejora y entrenamiento.

Si vemos estos procesos están alineados a diversas etapas del ciclo de vida del modelo de desarrollo de software sin perder de vista las características de la integración con el sistema operacional de la organización. (International Standard Organization, 2008)

Microsoft Security Development Lifecycle: es un modelo de la marca Microsoft que ayuda con desarrollo seguro y marcos de

cumplimiento está basado en 17 actividades mandatorias en el ciclo de vida y está basado en una estrategia de capacitación. (Microsoft, s.f.)

3.2 Ciclo de vida de las aplicaciones orientado al cumplimiento.

Un ciclo de vida de desarrollo de software con seguridad debe ser parte de una estrategia de empresa y de los entes reguladores y no solo debe contemplar los elementos de la construcción de los componentes a nivel de código, sino que debe contemplar la integración con la infraestructura tecnológica, el modelo de datos en su ciclo de vida, la integración de software y propiedad intelectual de terceros, el procesamiento de información de terceras partes, controles de cambios, administración de BUG, administración de vulnerabilidades, administración de incidentes, administración de requerimientos, continuidad, usabilidad, capacidades, respaldos, retención de información, así como muchos elementos que hoy en día se pueden prever como parte de un flujo de actividades que no son parte del proceso mismo del software, pero que hoy en día cobran gran relevancia.

Desde otra arista, las necesidades de la organización así como los recursos y las modalidades de operación pueden cambiar de proyecto a proyecto y con el paso del tiempo por lo que en ningún momento la forma en la que se defina que se operaran los proyectos de desarrollo de software deben tener injerencia en la definición de la seguridad, por lo que si se administra a través de proceso de cascada o bien por un modelo ágil no debe cambiar de ninguna forma el resultado final de la seguridad de la información.

Ahora desde la perspectiva legal es importante que todo modelo de desarrollo de software cumpla con las características que se encuentren alineadas a la protección de derechos de terceros como lo son la protección de la propiedad intelectual, protección de datos personales, seguridad de la información, uso transfronterizo de información, entre otros elementos.

En perspectiva de proponer un nuevo modelo de ciclo de vida, las fases deben estar alineadas al macroentorno y microentorno de la organización, por lo que a continuación se presentan los elementos que se creen necesarios para abordar todas las características previamente descritas:

Inicial: en esta actividad del ciclo de vida se deben obtener las características de amenazas del entorno, elementos normativos, así como la determinación del caso de negocio que conlleva la creación del proyecto de desarrollo de software, que debe estar alineada a un proceso de innovación en el cual se debe contemplar la cultura organizacional para la gestión del cambio.

Gobierno: en esta etapa se alinean todos los recursos y reglas de operación de las actividades de desarrollo, estrategias de contratación, estrategias de métricas y estrategias de capacitación y concientización de cada uno de los participantes en el proceso de desarrollo, para lo cual se determina un conjunto de políticas y procesos para el flujo constante y de revisión de cada uno de los parámetros de control.

Requerimientos: una vez que se detectaron las necesidades del macroentorno en materia de negocio para acreditar el caso de negocio, se deben establecer las características del mercado en términos de los requerimientos regulatorios, requerimientos de interacción con los clientes, requerimientos de interacción con los usuarios, asimismo se deben establecer las características funcionales y no funcionales de la operación del nuevo software, también en casos particulares se deben establecer las características de continuidad de la operación.

Arquitectura: en este punto es importante detectar las características de integración con la arquitectura de servicios de la empresa, posteriormente evaluar los componentes de arquitectura de tecnologías de la información así como del personal que se encuentra a cargo de la mismas con la finalidad de evaluar los componentes que se van a reutilizar y con ello también evaluar si se requerirá de proveedores de tecnologías adicionales y los cumplimiento normativos que estos deben llevar a cabo para su integración al modelo. Otro componente importante es validar el comportamiento del ciclo de vida de los datos para llevar a cabo un modelo de protección en cada una de las etapas. También en esta etapa se debe diseñar las pruebas de seguridad de la información que se deberán evaluar durante el proceso y con ello determinar los factores críticos de éxito de la seguridad del desarrollo.

Diseño: con base en los requerimientos y en la arquitectura, se realiza la composición del diseño con los elementos de seguridad que son apropiados y que se ajustan a los recursos y activos que se tienen previstos, así como el diseño de las modificaciones a los ciclos de vida de los datos y los procedimientos de negocio, en esta etapa se determinan los controles de seguridad que se deben llevar a cabo durante el proceso de despliegue y puesta en producción, en esta sección también se debe establecer los mecanismos de despliegue automatizado del código para evitar el traspaso de información entre los ambientes de desarrollo y pruebas. Se deben determinar las nuevas matrices o modificaciones de segregación de funciones.

Construcción: se deben seguir las reglas de desarrollo seguro, cuidando los elementos de debilidad y se debe probar el código de forma constante para su evaluación de forma

segmentada en consideración de las pruebas estáticas de código para evitar malas prácticas en el desarrollo de las aplicaciones, también se deben establecer métricas de desempeño para los equipos de desarrollo. Se debe contar con un repositorio con controles de versiones en los cuales se resguarde la documentación.

Pruebas: en esta actividad se deben llevar a cabo la comprobación de las pruebas de código estáticas de forma unitaria y validar las integraciones de la aplicación, adicionalmente se deben realizar análisis dinámicos de código y pruebas de penetración a las estructuras de información y con ello validar los ciclos de vida de los datos.

Despliegue y puesta en marcha: esto se debe realizar con un proceso de gestión de cambios, para lo cual se debe crear un cambio en los tipos de servicio en la mesa de servicios, modificar los modelos de segregación de funciones y la aplicación de estos a los perfiles de puesto, también se debe asegurar el endurecimiento de la seguridad de la información en la infraestructura tecnológica y por ende realizar el despliegue de forma automatizada.

Mantenimiento: en esta sección se debe mantener el proceso de gestión de incidentes, análisis de vulnerabilidades, pruebas de código constantes y pruebas de penetración.

Deposición: en esta última etapa del software se deben realizar los respaldos y los procesos de recuperación en caso de ser necesario, así como la definición de retención de la información.

Con base en este modelo propuesto de ciclo de vida de desarrollo de software no solamente se cumplen con las características técnicas, funcionales y de seguridad, sino que también cuenta con elementos que permiten delimitar las responsabilidades de cada uno de los participantes en el ciclo de vida del algoritmo desde su conceptualización hasta la destrucción de este y no dejando de lado que en cada una de las etapas se establecen nuevas relaciones con las partes involucradas.

Existen diversas metodologías que pueden apoyar en el ciclo de vida del desarrollo de los algoritmos, también se requiere que se estudien los modelos de cambios, ya que si bien pueden existir algoritmos que desde su creación hasta su destrucción no se modifiquen sus reglas de operación, es muy común que los cambios en las estructuras de los algoritmos se promuevan por cambios en el entorno o por la modificación del propósito de su creación, lo cual se puede ver desde la perspectiva de la creación de un nuevo algoritmo o bien solo la adecuación, lo cual en el ámbito jurídico puede trascender en la forma en la que impacta a los derechos de terceros o en el cumplimiento de las obligaciones, por lo que la gestión de cambios

en los algoritmos en este proceso deberá estar contemplado.

Conclusiones

Los algoritmos deben ser tratados como el conjunto de todos los elementos que interactúan para el fin especifico y los usos que se le pueden dar y no solo como el mecanismo de procesamiento de información, por lo que al estudiarlos en el ámbito jurídico y ético no solo se deben contemplar el código fuente o el código objeto, sino que se deben evaluar desde la conceptualización del procesamiento, el diseño, los modelos de despliegue y todo su ciclo de vida para que sea factible determinar el impacto que puede ejercer sobre la sociedad y también sobre las decisiones que toma de forma autónoma sin la intervención valorativa humana.

El ámbito regulatorio no solo debe estar limitado al algoritmo mismo sino también a la forma de conceptualización y la correspondiente responsabilidad de todos los sujetos que participan activa y pasivamente en el ciclo de vida, ya que también la omisión de las acciones puede generar consecuencias que afectan a derechos de terceros y no solo la ejecución del algoritmo.

También es importante estudiar las implicaciones jurídicas desde la perspectiva de los derechos humanos, la conducta en el uso y el propósito y muchos otros aspectos que no se han tocado en este documento pero que requieren de la atención debida.

Referencias

Black, P. E. (14 de 01 de 2009). *deterministic algorithm*. Obtenido de Dictionary of Algorithms and Data Structures: https://xlinux.nist.gov/dads/HTML/deterministicAlgorithm.html

Black, P. E. (28 de 02 de 2019). *randomized algorithm*. Obtenido de Dictionary of Algorithms and Data Structures [online]: https://xlinux.nist.gov/dads/HTML/randomizedAlgo.html

Black, P. E. (09 de 11 de 2020). *algorithm*. Obtenido de Dictionary of Algorithms and Data Structures [online]: https://xlinux.nist.gov/dads/HTML/algorithm.html

Black, P. E. (07 de 12 de 2020). *nondeterministic algorithm*. Obtenido de Dictionary of Algorithms and Data Structures: https://xlinux.nist.gov/dads/HTML/nondetermAlgo.html

Cámara de diputados del H. Congreso de la Unión. (21 de 12 de 2011). *Reglamento de la Ley Federal de Protección de Datos Personales en Posesión de los Particulares.* Obtenido de http://www.diputados.gob.mx/LeyesBiblio/regley/Reg_LFPDPPP.pdf

Cambridge. (s.f.). *Cambridge English Dictionary*. Obtenido de https://dictionary.cambridge.org/es-LA/dictionary/english/algorithm

Comisión Europea. (06 de 2018). *Directrices éticas para una IA confiable*. Obtenido de https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60423

Gonzalez, F., Ortiz, T., & Sanchez, R. (s.f.). *Uso responsable de la IA para las políticas públicas*. Obtenido de Inter-American Development Bank: https://publications.iadb.org/publications/spanish/document/IA-Responsable-Manual-tecnico-Ciclo-de-vida-de-la-inteligencia-artificial.pdf

International Standard Organization. (2008). *ISO/IEC 12207:2008.* Obtenido de https://www.iso.org/standard/43447.html

Microsoft. (s.f.). *What are the Microsoft SDL practices?* Obtenido de https://www.microsoft.com/en-us/securityengineering/sdl/practices

National Institute of Standars ans Technology. (10 de 2008). *Security Considerations in the System Development Life Cycle .* Obtenido de https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-64r2.pdf

Open Web Application Security Project. (s.f.). *Software Assurance Security Model*. Obtenido de https://owaspsamm.org/

Parlamento Europeo y Consejo de la Unión Europea. (17 de 04 de 2016). *Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo*. Obtenido de https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:32016R0679&from=EN

Payment Card Industry. (04 de 2021). *Secure Software Standard.* Obtenido de https://www.pcisecuritystandards.org/documents/PCI-Secure-Software-Standard-v1_1.pdf?agreement=true&time=1626203958571

Real Academia Española. (s.f.). *Diccionario de la lengua española*. Obtenido de https://dle.rae.es/algoritmo

Robotechnics. (11 de 11 de 2017). *Principios Asilomar de la Inteligencia Artificial*. Obtenido de https://www.robotechnics.es/asilomar/

The Open Web Application Security Project. (Noviembre de 2010). *Secure Coding Practices.* Obtenido de https://www.owasp.org/images/b/b2/OWASP_Development_Guide_2.0.1_Spanish.pdf

The Open Web Application Security Project. (Abril de 2017). *Estándar de Verificación de Seguridad en Aplicaciones*. Obtenido de https://www.owasp.org/images/a/aa/Est%C3%A1ndar_de_Verificaci%C3%B3n_de_Seguridad_en_Aplicaciones_3.0.1.pdf

Encuentra más artículos de la Revista Digital “Abogacía 4.0” en:
https://www.abogado.digital/revista-abogacia-40/