Sistemas y Computación

Systems and Computing

Autor: Laura Quintero Montoya
IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia
Correo-e: l.quintero2@utp.edu.co

***Resumen*— Este documento presenta un resumen de los principales contenidos del programa de Ingeniería de Sistemas y Computación. En el documento se explica el sentido de las cuatro grandes temáticas que se abordan en la carrera, y se indican sus principales aplicaciones en el campo industrial e investigativo. Las áreas son: programación, redes y comunicaciones, ingeniería de software e inteligencia artificial. El docente ha realizado la primera parte: programación, dejando para el estudiante la realización de los restantes tres temas: redes, software e inteligencia artificial.**

***Palabras clave*— sistemas, redes, inteligencia artificial, software, computación, investigación, industria.**

***Abstract*— This document presents a summary of the main contents of the Computer and Systems Engineering program. The document explains the meaning of the four major themes that are addressed in the career, and indicates their main applications in the industrial and research field. The areas are: programming, networks and communications, software engineering and artificial intelligence. The teacher has done the first part: programming, leaving the student to carry out the remaining three topics: networks, software and artificial intelligence.**

***Key Word*— systems, networks, artificial intelligence, software, computing, research, industry.**

I. INTRODUCCIÓN

El Programa Ingeniería de Sistemas y Computación estudia varios campos del conocimiento ligados a la teoría de la Informática y los Sistemas en general. Se han identificado varias áreas que representan el sustento teórico y práctico de la carrera, según se ha mencionado en el resumen del documento.

El objetivo del presente documento es describir cada uno de los temas mencionados, buscando con ello brindar una visión integral de la carrera, lo cual le permitirá al estudiante elegir aquellas temáticas que mejor se adapten a sus capacidades académicas.

I.1 PROGRAMACIÓN

En [1] se define la programación de la siguiente manera: "La programación informática es el proceso por medio del cual se diseña, codifica, limpia y protege el código fuente de programas computacionales. A través de la programación se dictan los pasos a seguir para la creación del código fuente de programas informáticos. De acuerdo con ellos el código se escribe, se prueba y se perfecciona."

Si se analiza la anterior definición, se aprecia que la programación se orienta a la solución de problemas técnicos y cotidianos a través de la escritura de un cierto código fuente, el cual debe respetar cierta estructura y método de trabajo. Para programar se debe conocer, con un buen grado de detalle, un lenguaje que se adapte al problema que se desea resolver.

Por ejemplo, si el problema a resolver es de carácter matemático, lo usual es que se emplee un lenguaje como Python, de gran acogida en los últimos tiempos. Una variante, más antigua pero igualmente importante, es el lenguaje Fortran, con el cual se desarrollaron las primeras soluciones a los problemas de Ingeniería.

Si el problema de tipo comercial, un lenguaje que se utilizó ampliamente es el lenguaje COBOL. Se dice que en la actualidad, y por un factor histórico, el 80% de las soluciones informáticas comerciales están elaboradas con este lenguaje.

Si la idea es resolver un problema de tipo general, se puede recurrir al lenguaje C, el cual se puede considerar como el padre de todos los lenguajes, pues fue utilizado en los orígenes de la computación moderna para el desarrollo del primer sistema operativo importante: UNIX.

Los lenguajes de programación se organizan según su modelo y estructura. A cada una de estas formas de organización se la conoce como: "Paradigma de Programación".

Según [2] un paradigma de programación es:

"Un paradigma de programación es un marco conceptual, un conjunto de ideas que describe una forma de entender la construcción de programa, como tal define:

- Las herramientas conceptuales que se pueden utilizar para construir un programa (objetos, relaciones, funciones, instrucciones).
- Las formas válidas de combinarlas.

Los distintos lenguajes de programación proveen implantaciones para las herramientas conceptuales descriptas por los paradigmas. Existen lenguajes que se concentran en las ideas de un único paradigma así como hay otros que permiten la combinación de ideas provenientes de distintos paradigmas.”.

Existen muchos paradigmas de programación. Los más importantes se describen a continuación:

PARADIGMA ESTRUCTURADO

El paradigma estructurado se basa en la ejecución secuencial y ordenada de instrucciones sobre un espacio de memoria debidamente organizada. Las estructuras básicas de programación son: secuencia, decisión y ciclo. Un lenguaje clásico de la programación estructurada es el lenguaje C.

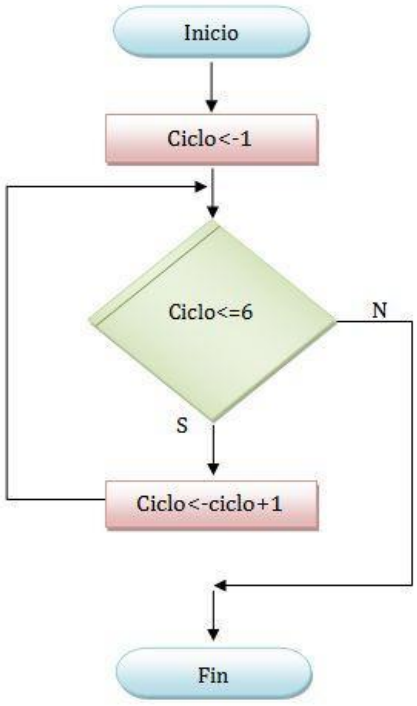

Figura 1. Paradigma estructurado

PARADIGMA DE OBJETOS

El paradigma de objetos es una concepción en la cual de definen entidades, denominadas clases, a partir de las cuales se crean objetos que interactúan entre sí. En cierto sentido, el paradigma de objetos es similar al concepto de objeto que se percibe en el mundo que nos rodea. Un lenguaje orientado a objetos es Smalltalk.

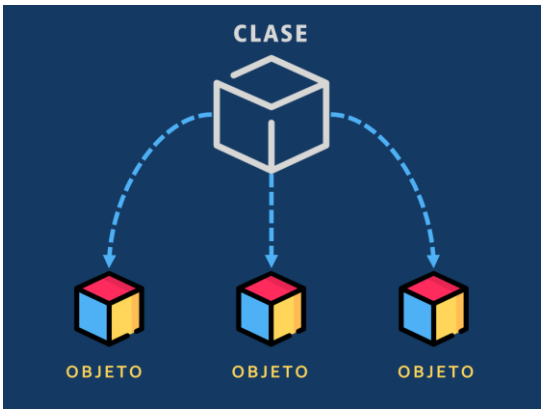

Figura 2. Paradigma orientado a objetos

PARADIGMA LÓGICO

El paradigma lógico está basado en la lógica de predicados de primer orden. Su objetivo es permitir extraer conclusiones a partir de premisas, de acuerdo con un conjunto de reglas y mecanismos de inferencia. Un lenguaje en el campo de la lógica es el PROLOG.

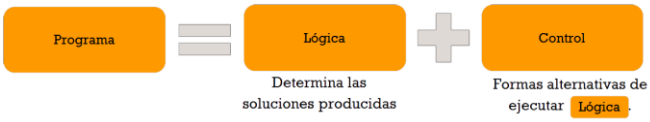

Figura 3. Paradigma lógico

PARADIGMA FUNCIONAL

El paradigma funcional se basa en la utilización de funciones como base de relación entre las partes de un programa. Una función es una porción de código que cumple un objetivo específico, permitiendo con ello simplificar y automatizar las tareas. Un lenguaje funcional es HASKELL.

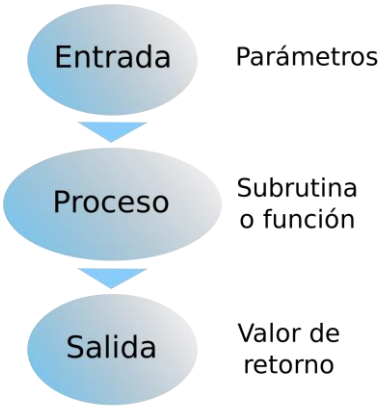

Figura 4. Paradigma funcional.

El paradigma estructurado se conoce, en ciertos entornos, como el paradigma IMPERATIVO. En la siguiente gráfica se aprecia lo visto hasta el momento:

Figura 5. Paradigmas de programación

Los paradigmas de programación, a su vez, se organizan en dos grandes categorías. La primera de ellas se conoce con el nombre de categoría IMPERATIVA. La segunda es la categoría DECLARATIVA.

La diferencia entre las dos categorías es la siguiente: en la categoría IMPERATIVA, los lenguajes de programación requieren que se indique de manera minuciosa cada uno de los pasos de la solución del problema. En este modelo se requiere realizar un seguimiento secuencial de cada paso a resolver en tal modelo.

En la categoría DECLARATIVA los lenguajes de programación no requieren de una descripción detallada y minuciosa de cada paso de la solución. Los lenguajes de tipo declarativo se caracterizan por disponer de un motor interno que les permite simplificar la ejecución de un programa. El motor le permite a los lenguajes encontrar caminos de solución que no están disponibles en el modelo imperativo.

En la siguiente gráfica se aprecia dicha clasificación.

Figura 6. Lenguajes imperativos y declarativos

Por último, se presenta un gráfico que presenta los principales lenguajes de programación.

Figura 7. Lenguajes de programación.

I.2 REDES Y COMUNICACIONES

De acuerdo con [3] las red se puntualiza como "Conjunto de computadoras o de equipos informáticos conect ados entre sí y que pueden intercambiar información." Y [4] que se basa la comunicación en "Transmisión de señales mediante un código común al emisor y al receptor."

Podemos ver esas dos palabras muy de la mano, es debido a su unificación en las tecnologías actuales, determinando estas sus avances y creación de nuevas herramientas, facilitando nuestro proceso de comunicación.

Las redes permiten que los programas de aplicación se comuniquen entre sí sin importar los sistemas operativos donde se ejecutan. La interconexión permite que los programas de aplicación se comuniquen independientemente de sus conexiones de red físicas.

Existen diversos tipos de redes, cada con sus diferentes características dependiendo de su población. Las redes pueden cubrir cualquier cosa, desde un puñado de dispositivos dentro de una sola habitación hasta millones de dispositivos repartidos por todo el mundo.

Por ejemplo, los email son vía de comunicación electrónica a larga distancia, sin importar su ciudad o región, el email se enviara de forma inmediata a su destinatario.

Algunas de las diferentes redes según el tamaño se pueden describir en:

RED DE ÁREA PERSONAL O PAN

Es una red de computadoras organizada alrededor de una persona individual dentro de un solo edificio. Esto podría ser dentro de una pequeña oficina o residencia.

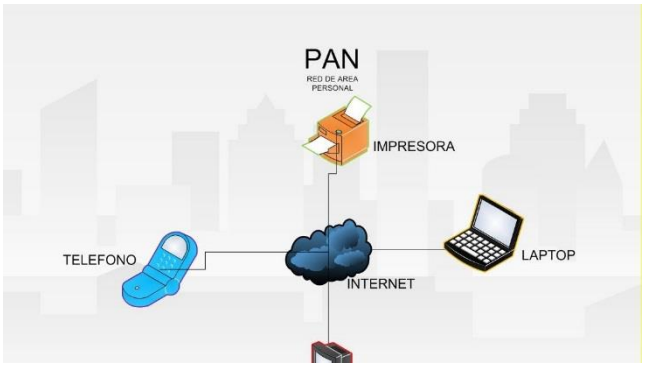

Figura 8. Redes de área personal

RED DE ÁREA LOCAL O LAN

Consiste en una red de computadoras en un solo sitio, generalmente un edificio de oficinas individual. Una LAN es muy útil para compartir recursos, como el almacenamiento de datos y las impresoras. Las LAN se utilizan normalmente para sitios únicos donde las personas necesitan compartir recursos entre ellos, pero no con el resto del mundo exterior.

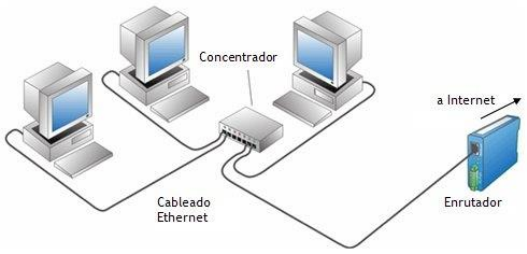

Figura 9. Redes de área local

RED DE ÁREA METROPOLITANA O MAN

Es una red informática en toda una ciudad, campus universitario o región pequeña. Un MAN es más grande que una LAN, que normalmente se limita a un solo edificio o sitio. Dependiendo de la configuración, este tipo de red puede cubrir un área desde varios kilómetros hasta decenas de kilómetros.

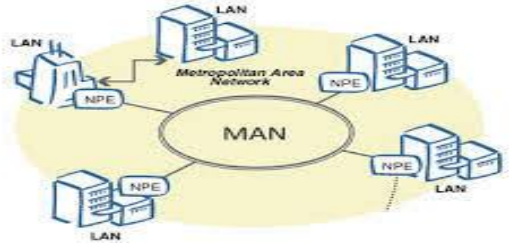

Figura 10. Redes metropolitanas

RED DE ÁREA AMPLIA O WAN

Ocupa un área muy grande, como un país entero o el mundo entero. Una WAN puede contener varias redes más pequeñas, como LAN o MAN. Internet es el ejemplo más conocido de WAN pública.

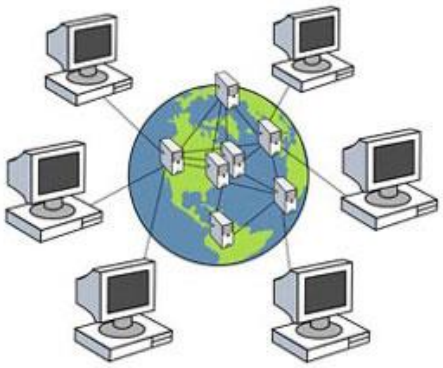

Figura 11. Redes metropolitana

Sin embargo, algunos tipos de redes tienen un propósito muy particular. Algunas de las diferentes redes en función de su finalidad principal son:

RED DE ÁREA DE ALMACENAMIENTO O SAN

Una red de área de almacenamiento (SAN) es una red especializada de alta velocidad que proporciona acceso de red a nivel de bloque al almacenamiento. Las SAN suelen estar compuestas por hosts, conmutadores, elementos de almacenamiento y dispositivos de almacenamiento que están interconectados mediante una variedad de tecnologías, topologías y protocolos. Las SAN también pueden abarcar varios sitios.

RED PRIVADA EMPRESARIAL O EPN

Cumple el propósito original de las redes privadas donde los datos están protegidos y los recursos se comparten. Las organizaciones construyen redes privadas empresariales para interconectar sus cuerpos dispares, es decir, los sitios de la empresa, como oficinas, sitios de producción, tiendas, almacenes, etc. con el fin de compartir recursos.

RED PRIVADA VIRTUAL O VPN

Es una conexión encriptado a través de Internet desde un dispositivo a una red. La conexión cifrada ayuda a garantizar que los datos confidenciales se transmitan de forma segura. Evita que personas no autorizadas detecten el tráfico y permite al usuario realizar el trabajo de forma remota.

I.3 INGENIERÍA DE SOFTWARE

En [5] la ingeniería de software se explica como “una disciplina formada por un conjunto de métodos, herramientas y técnicas

que se utilizan en el desarrollo de los programas informáticos (software).”

En referencia a la definición, La ingeniería de software se introdujo para abordar los problemas de los proyectos de software de baja calidad. Los problemas surgen cuando un software generalmente excede los plazos, el presupuesto y los niveles mínimos de calidad.

La ingeniera Se especializa en el estudio y desarrollo del software; y esta se divide en dos, sistemas informáticos y aplicaciones.

EL SOFTWARE DE SISTEMAS INFORMÁTICOS se compone de programas que incluyen herramientas informáticas y sistemas operativos. El SOFTWARE DE APLICACIONES consta de programas centrados en el usuario que incluyen navegadores web, programas de bases de datos, etc.

Los ingenieros de software tienen un amplio conocimiento de lenguajes de programación, desarrollo de software y sistemas operativos de computadora, y aplican principios de ingeniería a la creación de software. Al aplicar estos principios de ingeniería a cada etapa del proceso de desarrollo, desde el análisis de requisitos hasta el proceso de software, pueden crear sistemas personalizados para clientes individuales.

Así como un ingeniero civil se asegurará de que un puente tenga una base sólida, un ingeniero de software también comenzará con un estudio exhaustivo de los requisitos y trabajará en el proceso de desarrollo de manera sistemática.

En este momento, se está invirtiendo mucho en ingeniería de software debido a la creciente dependencia de la tecnología móvil, las empresas emergentes respaldadas por capital riesgo, la creciente complejidad de la tecnología y las industrias emergentes.

La demanda de ingenieros de software capacitados y calificados parece no tener fin. Esta demanda se ve reforzada por un panorama económico cambiante y alimentado por la necesidad de soluciones tecnológicas. Surgió para satisfacer la inmensa tasa de cambio en los requisitos del usuario y el entorno en el que se supone que funciona la aplicación.

OBJETIVOS

Los objetivos de la Ingeniería de Software son:

- la MANTENIBILIDAD, debería ser factible que el software evolucione para cumplir con los requisitos cambiantes.
- LA PORTABILIDAD, en este caso, el software se puede transferir de un sistema o entorno informático a otro.
- LA REUTILIZACIÓN, en ese caso un producto de software tiene una buena capacidad de reutilización si los diferentes módulos del producto pueden reutilizarse fácilmente para desarrollar nuevos productos.
- La ADAPTABILIDAD, en este tema, el software permite diferentes restricciones del sistema y las necesidades del usuario deben satisfacerse realizando cambios en el software.

RESPONSABILIDADES DE UN INGENIERO DE SOFTWARE

Las responsabilidades del ingeniero de software incluyen recopilar los requisitos del usuario, definir la funcionalidad del sistema y escribir código en varios lenguajes, como Java, Ruby on Rails o lenguajes de programación .NET (por ejemplo, C ++ o JScript.NET). Otras responsabilidades al mencionar son:

- Ejecutar el ciclo de vida completo del desarrollo de software (SDLC)
- Desarrollar diagramas de flujo, diseños y documentación para identificar requisitos y soluciones.
- Escriba código comprobable y bien diseñado
- Producir especificaciones y determinar la viabilidad operativa.
- Integre componentes de software en un sistema de software completamente funcional
- Desarrollar planes de verificación de software y procedimientos de garantía de calidad.
- Documentar y mantener la funcionalidad del software
- Solucionar problemas, depurar y actualizar sistemas existentes
- Implementar programas y evaluar los comentarios de los usuarios
- Cumplir con los planes del proyecto y los estándares de la industria.
- Asegúrese de que el software esté actualizado con las funciones más recientes.

Figura 12. Funciones de los ingenieros de software

PARADIGMAS DE SOFTWARE

En [6] para la Ingeniería de Software es "el paradigma es una agrupación de métodos, herramientas y procedimientos con el fin de describir un modelo."

Los paradigmas de software se refieren a los métodos y pasos que se toman al diseñar el software. Hay muchos métodos propuestos y están en funcionamiento en la actualidad, pero necesitamos ver en qué parte de la ingeniería de software se encuentran estos paradigmas. Estos se pueden combinar en varias categorías, aunque cada uno de ellos está contenido entre sí:

PARADIGMA DE DESARROLLO DE SOFTWARE

Este Paradigma se conoce como paradigmas de ingeniería de software donde se aplican todos los conceptos de ingeniería pertenecientes al desarrollo de software. Incluye varias investigaciones y recopilación de requisitos que ayudan a construir el producto de software. Consiste en:

- Recopilación de requisitos
- Diseño de software
- Programación

En el PARADIGMA DE DISEÑO DE SOFTWARE se desarrolla el diseño, mantenimiento y la programación, esta se encarga de producir un proceso y los pasos de su creación, mientras que el PARADIGMA DE PROGRAMACIÓN está estrechamente relacionado con el aspecto de programación del desarrollo de software, esta incluye codificación, pruebas e integración. En esta crea un programa que más adelante se va ejecutar, eso se adecua a la necesidad que se requerida.

Figura 13. Paradigmas del software

LA IMPORTANCIA DE LA INGENIERÍA DE SOFTWARE

La importancia de ingeniería de software surge debido a la mayor tasa de cambio en los requisitos del usuario y el entorno en el que funciona el software. Y para eso se necesita establecer estos parámetros:

SOFTWARE DE GRAN TAMAÑO

Es más fácil construir una pared que una casa o un edificio, del mismo modo, a medida que el tamaño del software se convierte en un gran tamaño, la ingeniería tiene que dar un paso para darle un proceso científico.

ESCALABILIDAD

Si el proceso de software no se basara en conceptos científicos y de ingeniería, sería más fácil volver a crear un nuevo software que escalar uno existente.

COSTO

A medida que la industria del hardware ha demostrado sus habilidades, la enorme fabricación ha reducido el precio del hardware informático y electrónico. Pero el costo del software sigue siendo alto si no se adapta el proceso adecuado. Todo eso dependiendo de su proyecto y su naturaleza de adaptación.

NATURALEZA DINÁMICA

La naturaleza siempre creciente y adaptable del software depende en gran medida del entorno en el que trabaja el usuario. Si la naturaleza del software siempre cambia, es necesario realizar nuevas mejoras en el existente. Aquí es donde la ingeniería de software juega un buen papel. Cada software se debe adaptar a los nuevos medios.

GESTIÓN DE LA CALIDAD:

Un mejor proceso de desarrollo de software proporciona un producto de software mejor y de calidad. No importa su costa, lo más importante es traer un proyecto con una buena eficacia.

I.4 INTELIGENCIA ARTIFICIAL

Según [7] la inteligencia artificial o la IA se define como "La Inteligencia artificial es el campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes"

Si se estudia la enunciación de la IA es una ciencia interdisciplinaria con múltiples enfoques, pero los avances en el aprendizaje automático y el aprendizaje profundo están creando un cambio de paradigma en prácticamente en todos los sectores de la industria tecnológica.

El objetivo expansivo de la inteligencia artificial ha dado lugar a muchas preguntas y debates. Tanto es así, que no se acepta universalmente una definición singular del campo.

La inteligencia artificial se basa en el principio de que la inteligencia humana se puede definir de manera que una máquina pueda imitarla fácilmente y ejecutar tareas, desde las más simples hasta las más complejas. Los objetivos de la inteligencia artificial incluyen el aprendizaje, el razonamiento y la percepción.

A medida que avanza la tecnología, los puntos de referencia anteriores que definían la inteligencia artificial se vuelven obsoletos. Por ejemplo, ya no se considera que las máquinas que calculan funciones básicas o reconocen texto a través del reconocimiento óptimo de caracteres incorporen inteligencia artificial, ya que esta función ahora se da por sentada como una función informática inherente. La IA evoluciona continuamente para beneficiar a muchas industrias diferentes. Las máquinas están conectadas utilizando un enfoque interdisciplinario basado en matemáticas, informática, lingüística, psicología y más.

Los algoritmos a menudo juegan un papel muy importante en la estructura de la inteligencia artificial, donde los algoritmos simples se utilizan en aplicaciones simples, mientras que los más complejos ayudan a enmarcar una inteligencia artificial fuerte.

APLICACIONES DE LA INTELIGENCIA ARTIFICIAL

Las aplicaciones de la inteligencia artificial son ilimitadas. La tecnología se puede aplicar a muchos sectores e industrias diferentes. La IA está siendo probada y utilizada en la industria de la salud para dosificar medicamentos y diferentes tratamientos en pacientes, y para procedimientos quirúrgicos en el quirófano.

También tiene aplicaciones en la industria financiera, donde se utiliza para detectar y marcar la actividad en la banca y las finanzas, como el uso inusual de tarjetas de débito y grandes depósitos en cuentas, todo lo cual ayuda al departamento de fraude de un banco. Las aplicaciones para IA también se están utilizando para ayudar a agilizar y facilitar el comercio. Esto se hace facilitando la estimación de la oferta, la demanda y el precio de los valores. En el caso de los vehículos autónomos, el sistema informático debe tener en cuenta todos los datos externos y calcularlos para que actúen de forma que se evite una colisión.

Figura 14. Aplicaciones de la inteligencia artificial

Los sistemas de inteligencia artificial fuertes son sistemas que llevan a cabo las tareas que se consideran similares a las humanas. Suelen ser sistemas más complejos y complicados. Están programados para manejar situaciones en las que se les puede requerir que resuelvan problemas sin que una persona intervenga. Este tipo de sistemas se pueden encontrar en aplicaciones como automóviles autónomos o en quirófanos de hospitales.

TIPOS DE INTELIGENCIA ARTIFICIAL:

La inteligencia artificial se puede dividir en dos tipos diferentes: DÉBIL y FUERTE. La inteligencia artificial DÉBIL incorpora un sistema diseñado para realizar un trabajo en particular. Los sistemas de inteligencia artificial débiles incluyen videojuegos como el ejemplo de ajedrez electrónico y asistentes personales como Alexa de Amazon y Siri de Apple. Le haces tarea como responder tareas o contestar preguntas al asistente, la responde por ti.

Figura 15. La IA débil, Alexa de Amazon

Los sistemas de inteligencia artificial FUERTES son aquellos sistemas que llevan a cabo las tareas mucho más complejo que simplemente contestar preguntar o realizar una tarea. Suelen ser sistemas más complejos y complicados. Están programados para manejar situaciones en las que se les puede requerir que resuelvan problemas sin que una persona intervenga. Un ejemplo de esto son los robots inteligentes como vector de anki y Sophia por Hanson Robotics.

Figura 16. La IA fuerte, Sophia la humanoide

LOS DESARROLLOS CLAVE EN IA

Todos estos avances han sido posibles gracias al enfoque en imitar los procesos de pensamiento humano. El campo de investigación que ha sido más fructífero en los últimos años es el que se conoce como “aprendizaje automático”. De hecho, se ha vuelto tan integral para la IA contemporánea que los términos "inteligencia artificial" y "aprendizaje automático" a veces se usan indistintamente.

Sin embargo, este es un uso impreciso del lenguaje, y la mejor manera de pensarlo es que el aprendizaje automático representa el estado actual de la técnica en el campo más amplio de la IA. La base del aprendizaje automático es que, en lugar de tener que aprender a hacer todo paso a paso, las máquinas, si pueden ser programadas para pensar como nosotros, pueden aprender a trabajar observando, clasificando y aprendiendo de sus errores, tal como lo hacemos nosotros.

EVOLUCIÓN DE LA INTELIGENCIA ARTIFICIAL

Los mayores avances en la investigación de la IA en los últimos años se han producido en el campo del aprendizaje automático, en particular en el campo del aprendizaje profundo.

Esto ha sido impulsado en parte por la fácil disponibilidad de datos, pero aún más por una explosión en el poder de cómputo paralelo en los últimos años, durante los cuales el uso de clústeres de GPU para entrenar sistemas de aprendizaje automático se ha vuelto más frecuente.

Estos clústeres no solo ofrecen sistemas mucho más potentes para entrenar modelos de aprendizaje automático, sino que ahora están ampliamente disponibles como servicios en la nube a través de Internet. Con el tiempo, las principales empresas de tecnología, como Google y Microsoft, han pasado a utilizar chips especializados adaptados tanto a los modelos de aprendizaje automático en ejecución como, más recientemente, a los de formación.

Un ejemplo de uno de estos chips personalizados es la Unidad de procesamiento de tensor (TPU) de Google, cuya última versión acelera la velocidad a la que los modelos útiles de aprendizaje automático creados con la biblioteca de software TensorFlow de Google pueden inferir información de los datos, así como la velocidad a la que pueden ser entrenados.

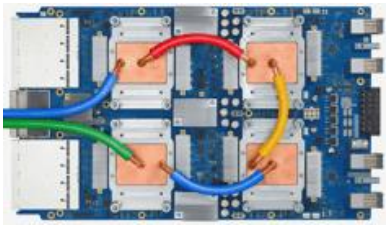

Figura 17. TPU de Google

USO ÉTICO DE LA INTELIGENCIA ARTIFICIAL

Si bien las herramientas de IA presentan una gama de nuevas funcionalidades para las empresas, el uso de la inteligencia artificial también plantea cuestiones éticas porque, para bien o para mal, un sistema de IA reforzará lo que ya ha aprendido.

Esto puede ser problemático porque los algoritmos de aprendizaje automático, que sustentan muchas de las herramientas de inteligencia artificial más avanzadas, son tan inteligentes como los datos que se brindan en el entrenamiento. Debido a que un ser humano selecciona qué datos se utilizan para entrenar un programa de IA, el potencial de sesgo de aprendizaje automático es inherente y debe monitorearse de cerca.

Cualquiera que busque utilizar el aprendizaje automático como parte de los sistemas de producción del mundo real debe tener en cuenta la ética en sus procesos de capacitación en inteligencia artificial y esforzarse por evitar sesgos. La aplicabilidad es un obstáculo potencial para el uso de IA en industrias que operan bajo estrictos requisitos de cumplimiento normativo.

COMPONENTES DE LA INTELIGENCIA ARTIFICIAL

A medida que se ha acelerado la exageración en torno a la IA, los proveedores se han esforzado por promover cómo sus productos y servicios utilizan la IA. A menudo, lo que ellos denominan IA es simplemente un componente de la IA, como el aprendizaje automático. La IA requiere una base de hardware y software especializados para escribir y entrenar algoritmos de aprendizaje automático. Ningún lenguaje de programación es sinónimo de IA, pero algunos, incluidos Python, R y Java, son populares.

REFERENCIAS

Referencias en la Web:

[1] https://conceptodefinicion.de/programacion-informatica/

[2] https://wiki.uqbar.org/wiki/articles/paradigma-de-programacion.html#:~:text=Un%20paradigma%20de%20programaci%C3%B3n%20es,relaciones%2C%20funciones%2C%20instrucciones).

[3] https://dle.rae.es/red#VXs6SD8

[4] https://dle.rae.es/comunicaci%C3%B3n?m=form

[5] https://definicion.de/ingenieria-de-software/

[6] https://helisulbaransistemas.blogspot.com/2014/09/paradigmas-en-el-desarrollo-de-software.html

[7] https://www.salesforce.com/mx/blog/2017/6/Que-es-la-inteligencia-artificial.html