

Sistemas y Computación

Systems and Computing

Autor: **José Gilberto Vargas Cano y Guillermo Andrés Sierra Rada**

IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: gilberto@utp.edu.co, guillermo.sierra@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.

1.1 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 es 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 descritas 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.

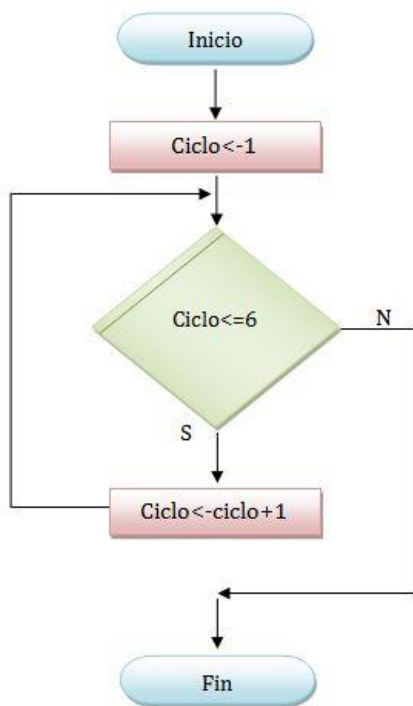


Figura 1. Paradigma estructurado

PARADIGMA DE OBJETOS

El paradigma de objetos es una concepción en la cual 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.

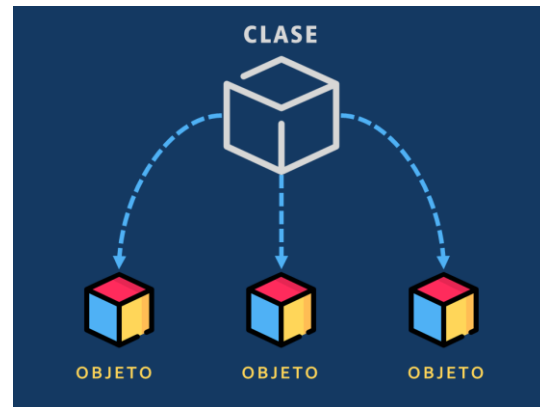


Figura 2. Paradigma orientado a objetos

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.

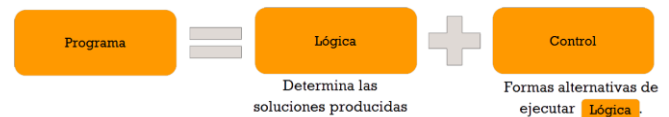


Figura 3. Paradigma lógico

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.

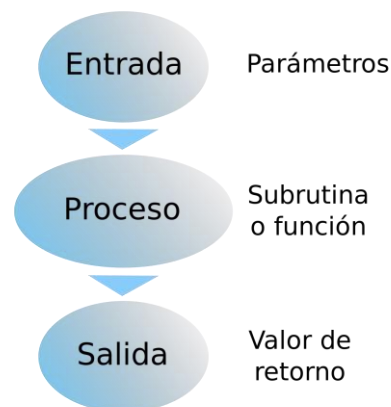


Figura 4. Paradigma funcional.

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.

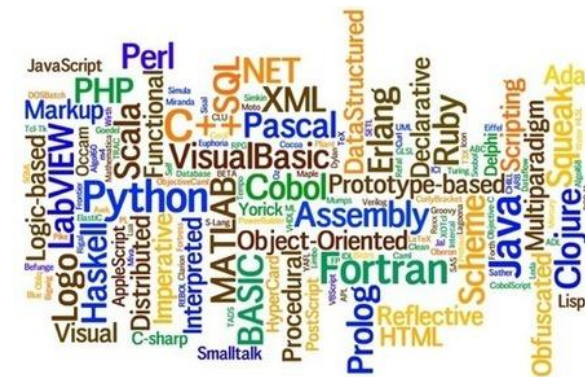


Figura 7. Lenguajes de programación.

1.2 REDES Y COMUNICACIONES

Según [3] “Se entiende por redes informáticas, redes de comunicaciones de datos o redes de computadoras a un número de sistemas informáticos conectados entre sí mediante una serie de dispositivos alámbricos o inalámbricos, gracias a los cuales pueden compartir información en paquetes de datos, transmitidos mediante impulsos eléctricos, ondas electromagnéticas o cualquier otro medio físico.”

Según lo anterior se puede observar la principal función de las redes de comunicaciones es mantener el contacto desde lejos y varía la forma de comunicar estas redes según como los usuarios necesiten usar las redes

Las redes pueden ser clasificadas según su alcance, método de relación y por relación funcional estas clasificaciones se pueden ver a continuación:

CLASIFICACIÓN SEGÚN SU ALCANCE

Se clasifican varias redes según el alcance en el que lleguen a brindar su señal.

PAN

Red de Área Personal o por sus siglas en inglés PAN es una red de computadoras para la utilización de los dispositivos de la computadora cerca de una persona, el alcance se una red PAN es de tan solo unos metros.

En la siguiente grafica se aprecia un ejemplo de red PAN



Figura 7. Red PAN

LAN

Red de Área Local o LAN por sus siglas en inglés, esta red es utilizada más por empresas se puede ver en aviones, edificios, un cuarto etc.... esta red se puede extender desde 200 metros hasta 1 kilómetro de distancia.

En la siguiente gráfica se presenta un ejemplo de red LAN

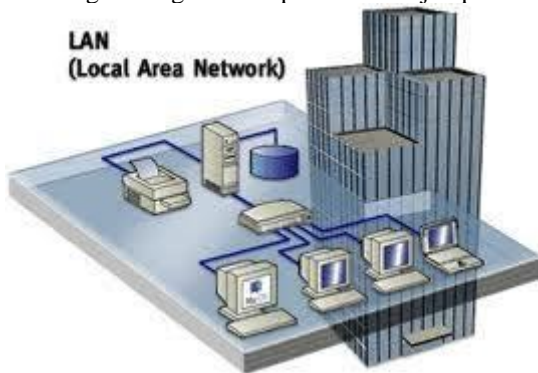


Figura 8. Red LAN

CAN

Red de Área de Campus o CAN por sus siglas en inglés se forma de varias redes LAN juntas que se instalan en un área geográfica específica como lo es un campus de universidad o un complejo industrial entre otros.

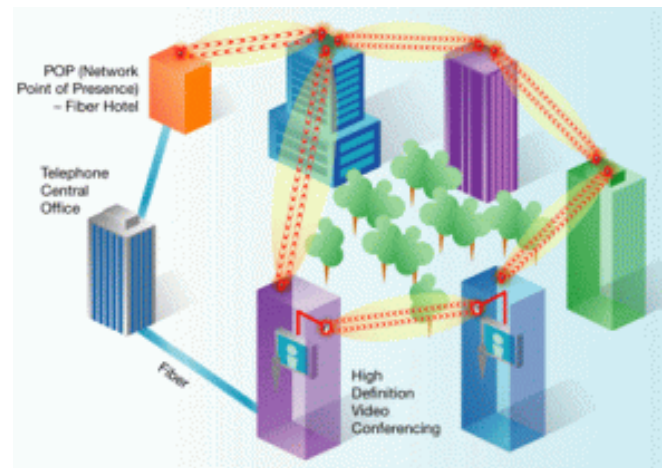


Figura 9. Red CAN

MAN

Red de Área Metropolitana o MAN por sus siglas en inglés es una Red que abarca un espacio que no sale de una ciudad y suele conectar con las redes LAN que hay a un alcancé de 50 kilómetros.

En la gráfica siguiente se presenta un esquema de como es una red MAN

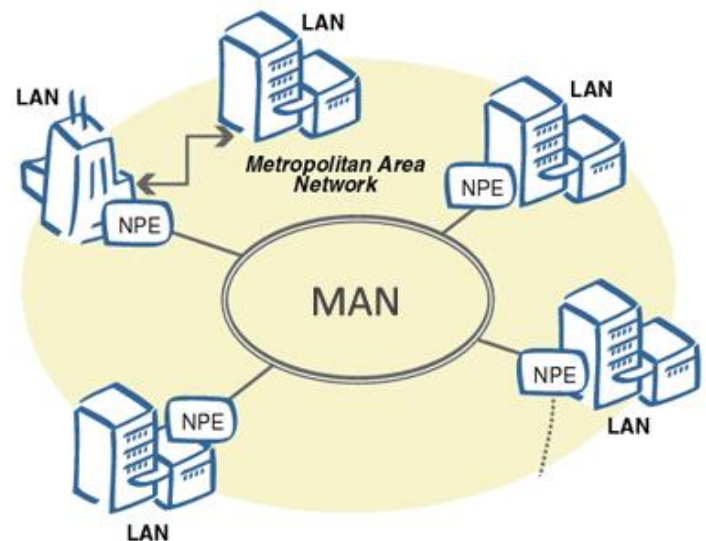


Figura 10. Red MAN

WAN

Red de Área Amplia o WAN por sus siglas en inglés es una red que cubre un espacio geográfico muy grande como una ciudad o un país se suele proveer de las empresas de internet.

En la siguiente gráfica se presenta un esquema de como es una red WAN

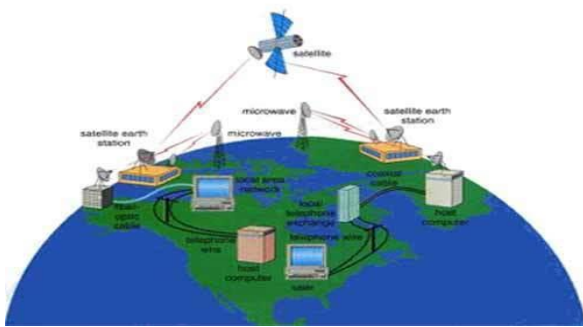


Figura 11. Red WAN

CLASIFICACIÓN SEGÚN SU MÉTODO DE CONEXIÓN

Hay redes que se clasifican según el método en el que envían la señal que deben.

MEDIOS GUIADOS

Los medios guiados son aquellos que transmiten su señal por un medio físico como lo son par trenzado, cable coaxial y fibra óptica.

En la siguiente gráfica se muestran los diferentes tipos de medios guiados.

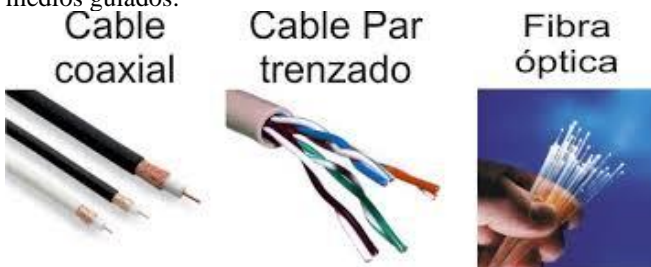


Figura 12. Medios Guiados

MEDIOS NO GUIADOS

Al contrario que los medios guiados este transmite su señal por medios no conectados o físicos entre estos están el radio, infrarrojos, bluetooth, satélite y láser.

En la siguiente gráfica podemos ver ejemplos de Medios no Guiados



Figura 12. Medios no guiados

TOPOLOGÍAS DE REDES

La topología de redes es la forma en como las redes están diseñadas para intercambiar datos, esto se expresa con figuras.

TOPOLOGÍA EN MALLA

En esta topología cada dispositivo se conecta directamente con los otros con esta topología los datos se pueden transmitir por varias vías.

En la siguiente gráfica se muestra un ejemplo de una red en malla.

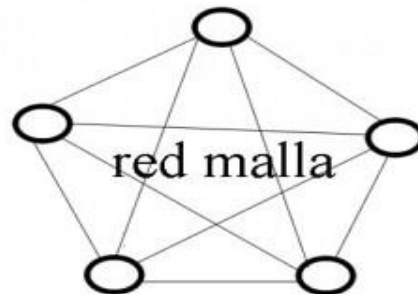


Figura 13. Red malla.

TOPOLOGÍA EN BUS O EN LÍNEA

En esta topología todos los dispositivos están conectados en la misma línea desde un cable, el cable procede de un dispositivo al siguiente y así sucesivamente.

En la siguiente gráfica se ve un ejemplo de red en bus.

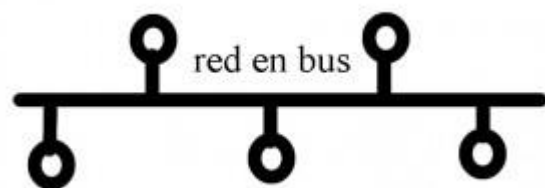


Figura 14. Red en bus

TOPOLOGÍA EN ESTRELLA

En la topología en estrella cada dispositivo está conectado a un solo controlador central que recibe y transmite todos los datos de los dispositivos.

En la siguiente gráfica podemos ver un ejemplo de red en estrella.

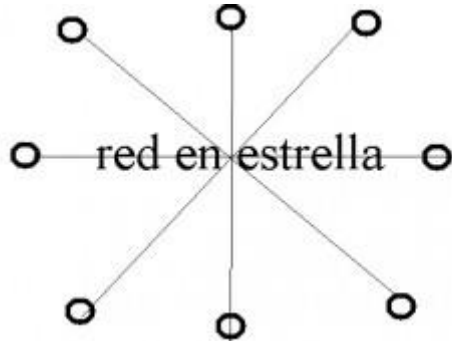


Figura 15. Red en estrella

TOPOLOGÍA EN ÁRBOL

Esta topología es parecida a una en estrella a diferencia que no está conectado por un controlador central, tiene varios controladores cada uno transmitiendo desde una red en estrella.

En la siguiente gráfica podemos ver un ejemplo de red en árbol.

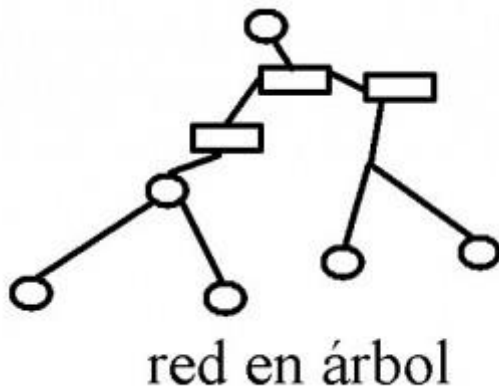


Figura 16. Red en árbol

TOPOLOGÍA EN ANILLO

En esta topología la conexión se da de un dispositivo al siguiente y este al siguiente así en círculo hasta que se llega al primer dispositivo; cada dispositivo del anillo incorpora un repetidor.

La siguiente gráfica podemos ver un ejemplo de red en anillo.

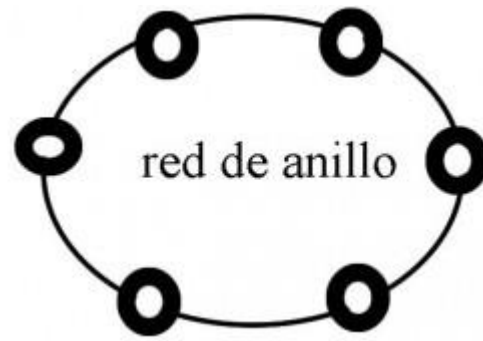


Figura 17. Red en anillo

1.3 INGENIERÍA DE SOFTWARE

Según [4] “La Ingeniería de Software es una de las ramas de las ciencias de la computación que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería. Brindando soporte operacional y de mantenimiento, el campo de estudio de la ingeniería de software.”

Según lo anterior se puede notar que los diferentes objetivos del software se basan en la modificación, mantenimiento y tratado de este para generar eficiencia en el uso del software, todo esto para generar software que tenga calidad como producto.

Dentro de los objetivos del software se miran muchos aspectos: los diseños de programas informáticos, el seguimiento de los costes del software, el crear un software confiable que no perjudique el sistema si hay un fallo, el buen mantenimiento del o los desarrolladores del software para mantenerse al tanto de la calidad del producto.

Dentro de la ingeniería del software se deben cumplir con ciertos requisitos para hacerse un buen ingeniero en este campo como lo son tener un conocimiento amplio en programación. La programación como se dijo es la empleación de un lenguaje con el que dictas que debe hacer un software que diseñas, otro requisito es seguir ciertas etapas o pasos por los que recorre un software en el orden adecuado; a continuación se mostrará las diferentes etapas o fases de la ingeniería del software.

ETAPA DE ANÁLISIS

En esta etapa se mira el problema o la necesidad que se tenga para identificar los requisitos de lo que será el desarrollo software adecuado, con la información recolectada a través de esta etapa se procede a la etapa de especificación.

ETAPA DE ESPECIFICACIÓN

En esta etapa se definen las libertades y restricciones de la funcionalidad del software a desarrollar, la operación y la interacción del software con los usuarios y los otros sistemas agregados y así con las especificaciones de esta etapa y la información de la anterior se puede proceder a la etapa de diseño.

ETAPA DE DISEÑO

En esta etapa se forma una estructura base de lo que se desarrollara luego teniendo en cuenta el análisis y las especificaciones del software que se debe realizar y considerando el hardware, la red entre otros para proseguir con la etapa de desarrollo.

ETAPA DE DESARROLLO

Durante esta etapa se elabora la programación y los implementos del software teniendo en cuenta el diseño que se definió en la etapa anterior. Una vez terminada esta etapa se puede proseguir con la etapa de prueba o validación.

ETAPA DE PRUEBA O VALIDACIÓN

En esta etapa se comprueba el software y sus especificaciones por cada punto y función que se le haya implementado una vez hechas estas comprobaciones se procede con la etapa de documentación.

ETAPA DE DOCUMENTACIÓN

Durante esta etapa se entrega el software junto con su respectiva documentación o manual con el propósito del futuro mantenimiento y/o actualizaciones, ya con esto hecho se prosigue con la etapa de mantenimiento.

ETAPA DE MANTENIMIENTO

Durante esta etapa se hace una inspección y mantenimiento del software para corregir errores y actualizar programas para mejorar la eficiencia y la confiabilidad del mismo así dando finalmente por desarrollado el software.

PRINCIPIOS DE UN INGENIERO DEL SOFTWARE

Una vez entendidas las facetas del desarrollo de un software apropiado para el usuario al mismo tiempo el ingeniero debe tener en cuenta ciertos principios a la hora de desarrollar, mantener o actualizar un software de esta manera mantener unos estándares para la calificación en cuando al servicio y el trabajo de un ingeniero del software.

Entre estos está: Ser dinámico pues un ingeniero necesita enfocarse en que lo que haga sea lo más fácil de manejar posible; centrarse en la calidad de su trabajo ya que un

ingeniero siempre debe estar seguro de que tan bueno es su trabajo; adaptabilidad, durante el desarrollo o mantenimiento de un software pueden haber imprevistos bruscos que fuercen al ingeniero a cambiar su forma de trabajar; comunicación efectiva, esta se necesita sobre todo cuando se habla con un cliente o con un compañero de trabajo al cual le quieras expresar tus ideas y observaciones; administrar el cambio, esto es algo importante a la hora de que se tenga que modificar el trabajo por ciertas complicaciones o necesidades que aparezcan con el tiempo de esta manera obtener una calidad de trabajo optima; calcular el riesgo puede ser vital a la hora de desarrollar un software para evitar que pequeños detalles puedan arruinar el trabajo o en caso de que esto ocurra tener un plan de emergencia para mantener el progreso estable y finalmente elaborar un trabajo que colabore en la comunidad y que este libre a modificaciones, esto es fundamental para que entre ingenieros se puedan implementar cambios importantes, recordar que los conocimientos que se adquieran desarrollando un trabajo pueden servir a otros para desarrollar su propio trabajo.

El tener etapas y requisitos anteriormente aclarados ayuda a saber diferenciar entre un malo y excelente ingeniero del software.

1.4 INTELIGENCIA ARTIFICIAL

Según [5] “La inteligencia artificial (IA) es la inteligencia llevada a cabo por máquinas. En ciencias de la computación, una máquina «inteligente» ideal es un agente flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea.”

Según lo anterior podemos definir en palabras más simples la inteligencia artificial es una ampliación de la tecnología que se diseñó para asemejar lo más posible la funcionalidad del cerebro humano. La IA es actualmente la máxima expresión de la automatización.

La inteligencia artificial a pesar de las diferentes formas que puede tener tiene características que siempre perduran en cada una de estas formas, estas características son importantes para que la inteligencia artificial se pueda considerar eficiente, entre estas están: el aprendizaje el cual es importante para poder recolectar la información necesaria para actuar con el razonamiento adecuado, el razonamiento es otra característica la cual una IA utiliza para llegar a conclusiones lógicas del problema que deba resolver y la autocorrección, la siguiente característica la cual en uso de las anteriores cambia aspectos en su comportamiento para mejorar su trabajo.

La inteligencia artificial es desarrollada por programadores, ingenieros de software y científicos de otros campos, todos estos desarrolladores necesitan tener en cuenta ciertos principios que deben seguir para desarrollar una IA, para esto se desarrollo el termino F.A.T.E, un acrónimo en inglés que

representa unas características que deben tener en cuenta los desarrolladores para poder crear inteligencia artificial que pueda manejarse y adaptarse a una sociedad sana. Como acrónimo cada una de estas letras representa una caracterización importante, las cuales son:

Fairness o justicia, toda tecnología debe estar pensada para que no discrimine a nadie, garantizando la igualdad a cualquiera que la use o quiera usar

Accountability o responsabilidad, el uso de la tecnología artificial puede tener ciertas consecuencias y cada persona que utilicé o desarrollé una IA debe responder por las consecuencias de esta

Transparency o transparencia, la inteligencia artificial debe explicarse de una forma simple o al menos entendible, que no se explique en términos no técnicos es importante para que sea comprensible por una persona promedio la cual quiera saber que y como utilizar una IA.

Ethics o ética, cada desarrollador y persona que utilice una IA tiene que verificar que las funciones y el uso de este estén de acuerdo a los valores básicos que como sociedad aceptamos.

TIPOS DE INTELIGENCIA ARTIFICIAL

Entre las IA existen un tipado que se ejerce según su funcionamiento, como se maneja, para que trabajos específicos se emplean entre otras muchas características.

A continuación, se mencionarán los 4 tipos de IA con su respectiva información.

MÁQUINAS REACTIVAS

Es el tipo más básico de IA, solo tiene en cuenta el proceso presente, este tipo de IA ve un problema que le pongan en frente, lo resuelve y se olvida de él por completo.

MEMORIA LIMITADA

Este tipo de IA puede guardar información del pasado con duración limitada, hace uso de esta información para modificar algo del proceso ya programado de antes así poder dar un funcionamiento algo más “personal” al tipo de situación en el que se encuentre, aún así esta IA no puede guardar información por mucho tiempo como lo hace el cerebro humano.

TEORÍA DE LA MENTE

La teoría de la mente se basa en las personas, en este caso aplicado a las máquinas la función principal es deducir los sentimientos e intenciones de un humano y reaccionar de acuerdo a esto. Un acercamiento muy personal a la mente humana.

AUTOCONCIENCIA

Este tipo de IA aún no es visto actualmente y es que representa el máximo de una máquina y en sí el máximo de nuestra especie, el que una máquina pueda ser autoconsciente significa que puede interiorizarse, a sus pensamientos y conllevar una teoría de la mente sin problemas y estar a la altura de un humano o superarlo por sus capacidades como máquina.

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\).](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://concepto.de/redes-informaticas/#ixzz6VbKX3iUA>
- [4] https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software
- [5] https://es.wikipedia.org/wiki/Inteligencia_artificial