Sistemas y Computación

Systems and Computing

Autor: Steveen Restrepo Corrales

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

Correo-e: s.restrepo3@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.**

1. 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. 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.

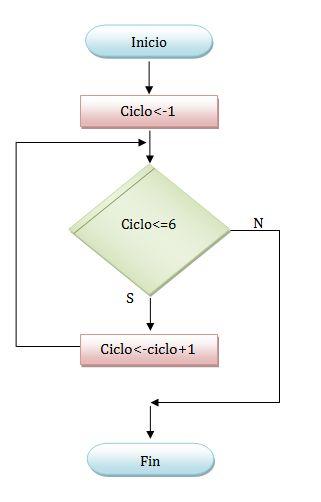


Figura 1. Paradigma estructurado

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.

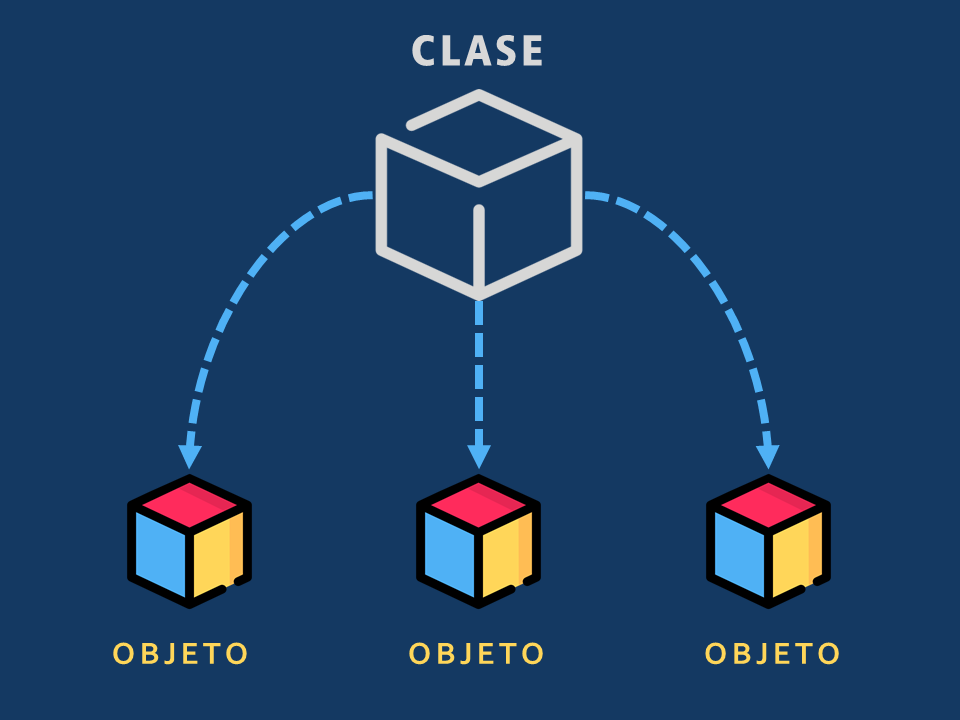


Figura 2. Paradigma orientado a objetos

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.

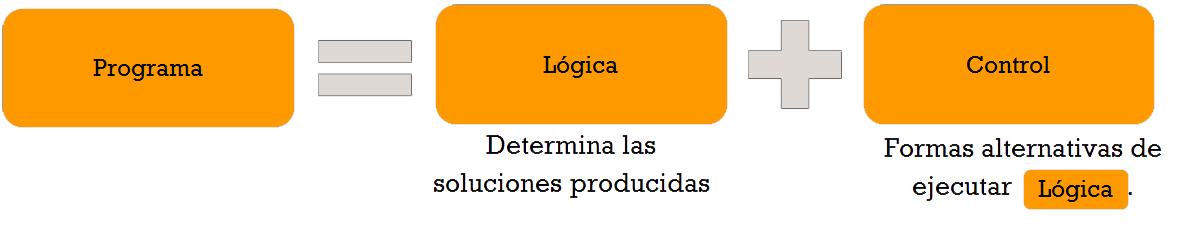


Figura 3. Paradigma lógico

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.

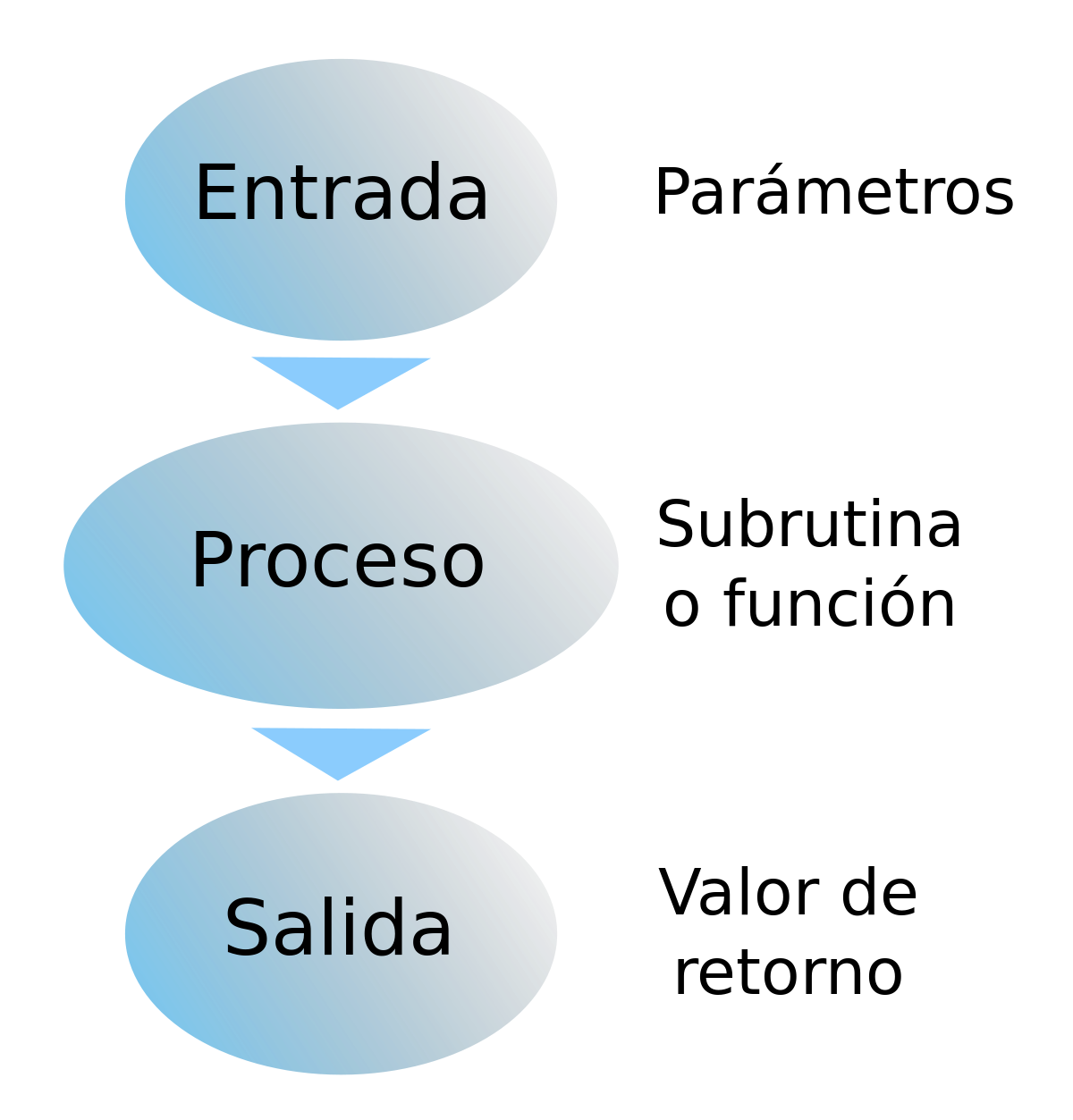


Figura 4. Paradigma funcional.

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.

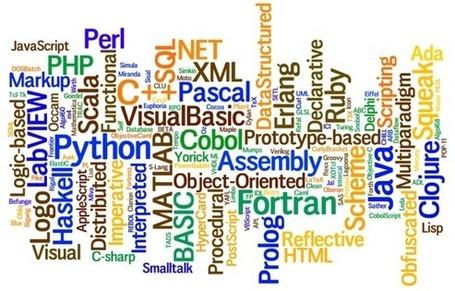


Figura 7. Lenguajes de programación.

* 1. REDES Y COMUNICACIONES

Según [3] una red es la combinación de dos o mas sistemas y los enlaces de conexión de estos. Al hardware se le considera una red fisca (equipos como adaptadores, cables y líneas de teléfono) que componen la red. El software y el modelo conceptual componen la red lógica. Existen distintos tipos de redes y emuladores que proporcionan funciones diferentes.

La complejidad de las redes de sistemas modernas ha dado origen a varios para explicar el funcionamiento de las redes. Entre ellos uno de los más comunes es el Modelo de referencia OSI (Open Systems Interconnection – Interconexión de sistemas abiertos) de la International Standards Organization (Organización internacional para los estándares), que también se conoce como el modelo de las 7 capas OSI.

Las 7 capas del modelo OSI se describen del siguiente modo:

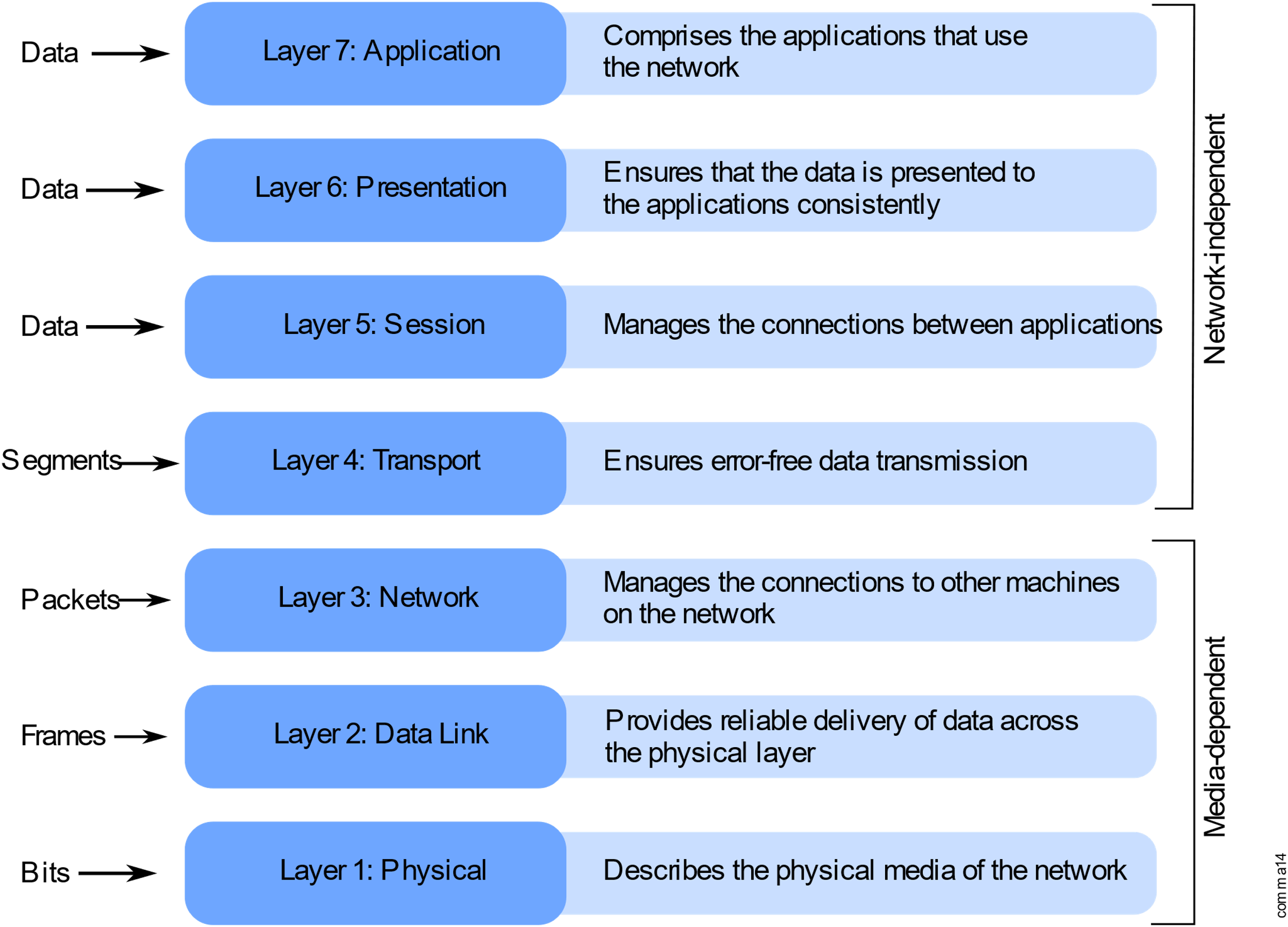


Figura 7. Modelo de referencia OSI.

* **Transferir datos.** Se puede transferir datos de un sistema a otro. Es posible migrar archivos, directorios y sistemas de archivos completos de una maquina a otra a través de la red, lo que permite realizar copias de seguridad remotas y garantiza la redundancia en caso de que se produzca una anomalía en la máquina.
* **Entrada de datos.** La entrada de datos consiste en entrar datos directamente en los archivos de datos locales o remotos. El incremento de precisión y de eficiencia es la consecuencia natural de una transferencia de datos de un solo paso.
* **Comunicaciones con otros sistemas operativos.** Una red puede tener conectados distintos tipos de sistemas. Los sistemas pueden ser fabricantes diferentes o ser modelos diferentes del mismo fabricante. Los programas de comunicaciones subsanan las diferencias entre los sistemas operativos de dos o más tipos de sistemas.
  1. INGENIERÍA DE SOFTWARE

[4] La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos, más conocidos como software.

Este servicio trasciende a la programación, que es la base para crear una aplicación. La ingeniería de software engloba toda la gestión de un proyecto. Desde el análisis previo de la situación, el planteamiento del diseño hasta su implementación, pasando por las pruebas recurrentes para su correcto funcionamiento.

**Etapas de la ingeniería de software**

Dentro de la ingeniería de software entendemos que también se encuentra todo el proceso de elaboración del software, que se denomina ciclo de vida. Está formado por cuatro etapas:

* **Concepción.** En esta primera fase se desarrolla el modelo de negocios. Es decir, conocemos las necesidades que debe de tener un software y empezamos a buscar las herramientas para cubrirlas.
* **Elaboración.** Se detalla las características de la estructura del software.
* **Construcción.** Tal y como su nombre indica en este paso empezaremos a elaborar de forma tangible todo aquello que, de momento, solo hemos plasmado en formas de ideas.
* **Transición.** Es el momento de la implementación y el desarrollo para los clientes o usuarios. Deben tener tiempo para familiarizarse con el nuevo software

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 8. Interfaz de usuario de grafica.

**Objetivos de la ingeniería en software**

* Diseños de programas informáticos adaptados a las necesidades y exigencias de los clientes.
* Estar presente en todas las fases del ciclo de vida de un producto.
* Contabilizar los costes de un proyecto y evaluar los tiempos de desarrollo.
* Realizar el seguimiento del presupuesto y cumplir los plazos de entrega.
* Diseñar, construir y administrar bases de datos.
  1. INTELIGENCIA ARTIFICIAL

[5] La inteligencia artificial (IA) es lo que hace posible que las maquinas aprendan de la experiencia, también que se ajusten a nuevas aportaciones y realicen tareas como seres humanos. Hoy en día ya se oye hablar sobre ejemplos de la inteligencia artificial – desde computadoras que juegan ajedrez hasta vehículos que se conducen de forma autónoma – recurren mayormente al aprendizaje profundo y al procesamiento del lenguaje natural. Al emplear estas tecnologías, se puede entrenas a las computadoras para realizar tareas especificas procesando grandes cantidades de datos y reconociendo patrones de datos.

**Historia de la inteligencia artificial**

Este término se empezó a utilizar en el año 1956, pero se ha vuelto más popular hoy en día gracias al incremento en los volúmenes de datos, algoritmos avanzados y mejoras en el poder de cómputo y aprendizaje.

Esta tuvo sus inicios en la década de 1950 en la cual exploraba temas como la solución de problemas y métodos simbólicos. En la década de 1960 el Departamento de Defensa de los Estados Unidos mostro interés en este tipo de tecnología y comenzó a entrenar computadoras para que imitaran el razonamiento humano básico. Además de esto, la Defense Advanced Research Project Agency (DARPA, Agencia de Proyectos de Investigación Avanzada de Defensa) realizo proyectos de planimetría en la calle en la década de 1970 y DARPA también produjo asistentes personales inteligentes en 2003, mucho antes de que Siri, Alexa o Cortana fueran conocidas.

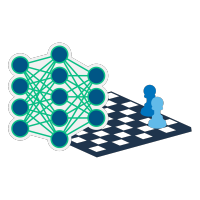


Figura 9. 1950-1970 Redes neuronales

Icono

Descripción generada automáticamente

Figura 9. 1980-2010 Aprendizaje Automático

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 10. Presente Deep Learning

**¿Por qué es importante la inteligencia artificial?**

* La inteligencia artificial automatiza el aprendizaje y descubrimiento a través de datos.
* La inteligencia artificial analiza mas datos y datos más profundos empleando redes neuronales que tienen muchas capas ocultas.
* La inteligencia artificial logra una increíble precisión a través de redes neuronales profundas- lo cual antes era imposible.
* La inteligencia artificial saca el mayor provecho de los datos
* La inteligencia artificial se adapta a través de algoritmos de aprendizaje progresivo para permitir que los datos realicen la programación.

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://www.ibm.com/docs/es/aix/7.2?topic=management-network-communication-concepts>

[4] <https://systemsgroup.es/tecnologias-de-la-informacion/la-ingenieria-de-software-que-es-y-que-utilidad-tiene/32363/>

[5] <https://www.sas.com/es_co/insights/analytics/what-is-artificial-intelligence.html>