

# Sistemas y Computación

## Systems and Computing

Autor: Natalia Alzate Usuga

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

Correo-e: Natalia.alzate@utp.edu.co

**Resumen—** Este documento se 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 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.

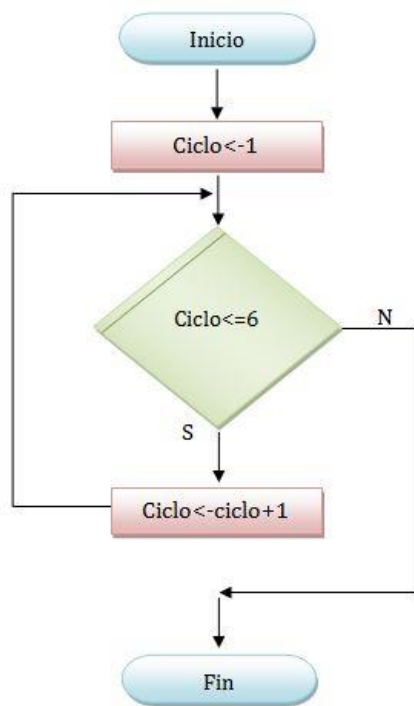


Figura 1. Paradigma estructurado

### PARADIGMA DE OBJETOS

El paradigma de objetos es una concepción en la cual se 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.

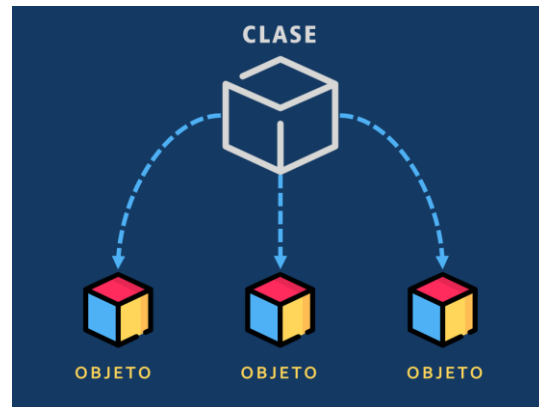


Figura 2. Paradigma orientado a objetos

### 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.

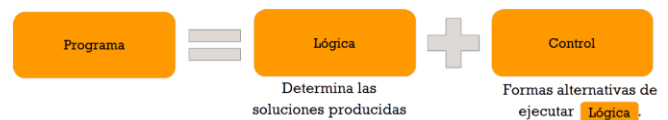


Figura 3. Paradigma lógico

### 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.

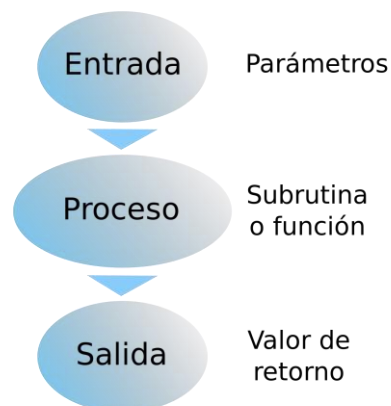


Figura 4. Paradigma funcional.

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.

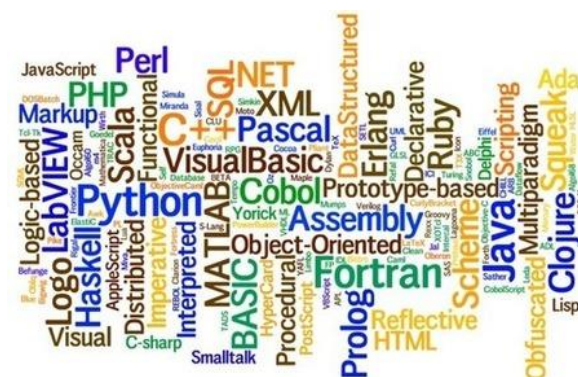


Figura 7. Lenguajes de programación.

## 1.2 REDES Y COMUNICACIONES

Una red se compone generalmente de dos o más sistemas y de enlaces de conexión. Es importante tener en cuenta dos conceptos claves que nos ayudarán a comprender mejor la composición de una red:

1. Hardware, es la parte física de la red. Un claro ejemplo serían los cables, la computadora en sí, el modem entre otros.
2. Software, es la parte intangible de la red. Ésta compone la parte lógica.

Según [3] “La complejidad de las redes de sistemas modernas ha dado origen a varios modelos conceptuales para explicar cómo funcionan las redes. Entre estos modelos, 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 modelo de siete capas OSI.”

En la siguiente gráfica encontraremos cada una de las capas y sus protocolos asignados

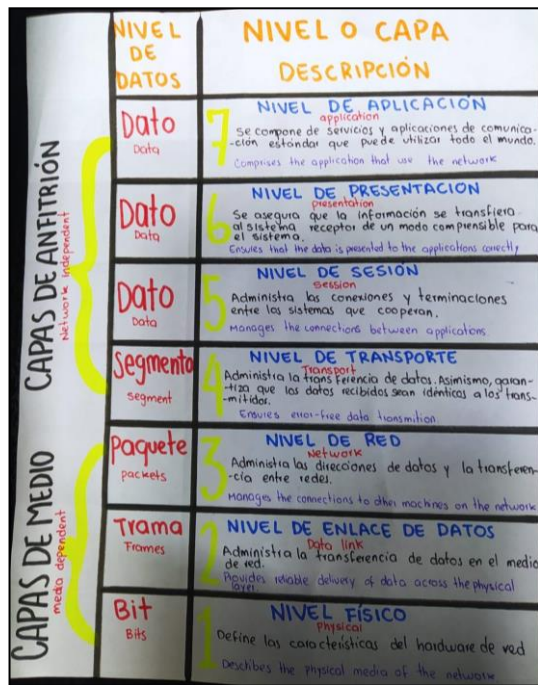


Figura 8. Modelo OSI.

A su vez, las redes nos permiten diversas funciones encaminadas a las comunicaciones tanto de aplicaciones como usuarios, algunos ejemplos son: Enviar correos electrónicos, transferir datos, consultar datos, realizar funciones remotas, compartir datos y recursos, entre otros.

Existen distintos tipos de redes (compuestas por hardware y software) y emuladores (Una aplicación de software):

1. Redes físicas, usualmente cables coaxiales, par trenzado, fibra óptica o líneas telefónicas que conectan un hardware.
2. Sistema de red, es la utilización del hardware y el software al unísono para lograr intercambiar información con la red.
3. Comunicación con otros sistemas operativos, permite la conexión entre diferentes tipos de sistemas, siendo éstos de diferentes modelos y/o fabricantes.
4. Aplicaciones de emulación de sistema principal, es la habilidad de utilizar el software de manera remota. Un ejemplo podría ser el programa TeamViewer.
5. Mandatos de sistema de comunicaciones, Describe las ordenes que se pueden utilizar para identificar diferentes usuarios del sistema, permitiendo una personalización de perfil.

### 1.3 INGENIERÍA DE SOFTWARE

Basándonos en la clara definición de [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. (...) incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema.”

El término “Ingeniería de software” surge cuando se llevaba a cabo dos conferencias estructuradas por la OTAN en los años en 1967 y 1968. A partir de allí podemos marcar una diferencia entre la ingeniería de software y la ingeniería de sistemas. La segunda solo se encarga de crear y aplicar técnicas de información. Mientras la primera se encarga de las funciones listadas en la siguiente gráfica.

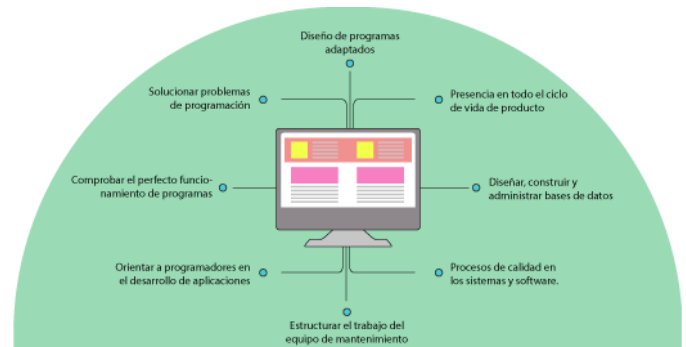


Figura 9. Campo de la ingeniería de software.

Para llevar a cabo el desarrollo de un software se sigue una serie de pasos que se conoce como “Ciclo de vida” que tiene en total 7 etapas:

1. Análisis
2. Diseño
3. Desarrollo
4. Pruebas y/o verificación
5. Implementación
6. Mantenimiento
7. Final o EOL (End Of Life)

### 1.4 INTELIGENCIA ARTIFICIAL

Se entiende como la combinación de algoritmos y software, para brindar a máquinas la capacidad de imitar la mente humana y así poder pensar, razonar y actuar, aprendiendo constantemente de la experiencia. Tenemos programas informáticos que reciben órdenes y desarrollan soluciones basados en reglas previamente establecidas, diferente a la inteligencia artificial donde se llegan a soluciones sin recibir

órdenes de un programa y cubriendo una gran variedad de escenarios.

Un concepto importante en la AI (Inteligencia Artificial) es Aprendizaje Automático (Machine Learning), que se logra tras tener un aprendizaje inicial, entrenar ese aprendizaje, lo que creará una experiencia y por último se lleva a la práctica.

Este aprendizaje se divide en dos tipos: **Supervisado**, donde un humano puede corregirlo y **No Supervisado**, donde la máquina tiene que valerse por ella sola y sus conocimientos adquiridos

También se intenta imitar las redes neuronales para copiar el funcionamiento del cerebro y los enlaces y transmisión de información. En [5] se explica como “Una **neurona artificial** es una entidad que recibe unos datos de entrada, les aplica una serie de operaciones matemáticas y una función de activación (una fórmula matemática), y genera un resultado. Es un mecanismo sencillo, pero la complejidad llega cuando millones de neuronas trabajan en paralelo para crear Redes Neuronales Artificiales, o RNA.”

Existen diferentes tipos de inteligencia artificial:

- Sistemas que piensan como humanos, esto quiere decir que se automatiza la toma de decisiones, la resolución de problemas y el aprendizaje, gracias a la implementación de las redes neuronales artificiales.
- Sistemas que actúan como humanos, el caso más familiar es un robot que realiza tareas cotidianas como una persona lo haría.
- Sistemas que piensan racionalmente, es la capacidad que tiene la inteligencia artificial para actuar basándose en la lógica.
- Sistemas que actúan racionalmente, intentan imitar el raciocinio humano, es decir que analizan la información.

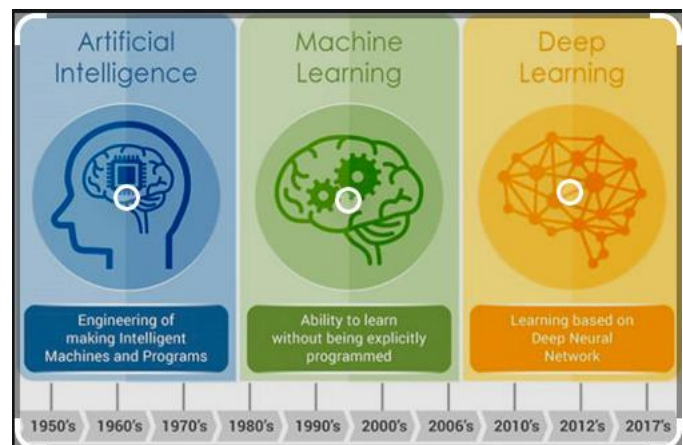


Figura 10. Desarrollo de la IA a través del tiempo.

## 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://www.ibm.com/docs/es/aix/7.2?topic=management-network-communication-concepts>
- [4] <https://definicion.de/ingenieria-de-software//>
- [5] <https://computerhoy.com/reportajes/tecnologia/inteligencia-artificial-469917>