

# Sistemas y Computación

## Systems and Computing

Autor: **Yohalinson Omar Montoya Plaza**

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

Correo-e: [yohalinson.montoya@utp.edu.co](mailto:yohalinson.montoya@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 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.

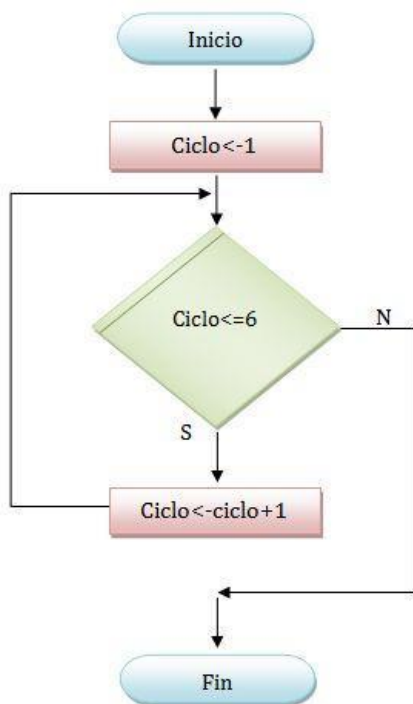


Figura 1. Paradigma estructurado

### 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.

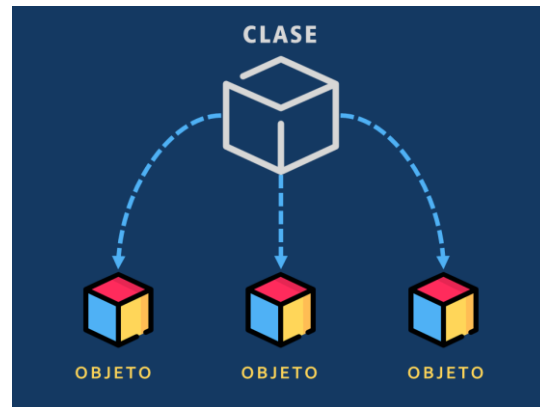


Figura 2. Paradigma orientado a objetos

### 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.

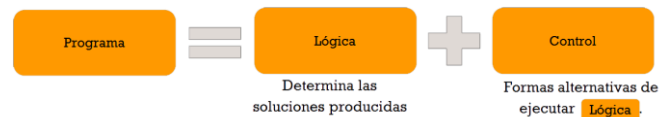


Figura 3. Paradigma lógico

### 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.

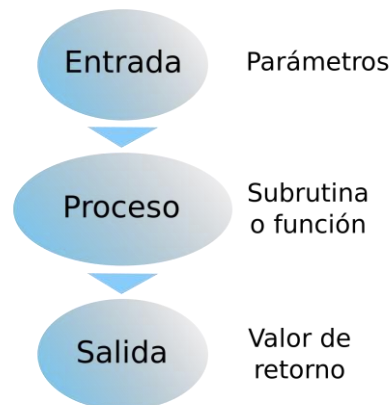


Figura 4. Paradigma funcional.



ejemplo, un navegador de Internet) y un servidor. La información que se transmite en este proceso recibe el nombre de **recurso**, identificado a través de un URL (*Localizador Uniforme de Recursos*).

## FTP

El *Protocolo de Transferencia de Archivos*, por su parte, se utiliza cuando se desea enviar y recibir archivos de un sistema a otro, siempre que ambos se basen en la arquitectura *cliente-servidor* y que se encuentren conectados a una red que cumpla con el **TCP**, explicado en la definición de **protocolo de comunicación**. El FTP permite que un usuario se conecte a un servidor para bajar archivos o bien para subirlos, sin la necesidad de que ambos equipos utilicen el mismo sistema operativo.

## SMTP

Con un nombre menos conocido que los dos anteriores, el *Protocolo para transferencia simple de correo* es utilizado una cantidad incalculable de veces al día por usuarios de todo el mundo, ya que da forma al intercambio de mensajes de **correo electrónico** (también conocido como *e-mail* o *email*) entre una amplia gama de dispositivos, como ser los teléfonos móviles, las tabletas y los ordenadores. Se trata de un estándar oficial cuya operación se encuentra en manos de los proveedores de servicios de email.

## POP

El *Protocolo de Oficina de Correo* o de *Oficina Postal* brinda a los usuarios la posibilidad de recibir y almacenar el correo electrónico en un **equipo** local. En la actualidad se prefiere el uso de POP3, la versión más reciente, dado que las primeras dos se consideran obsoletas.

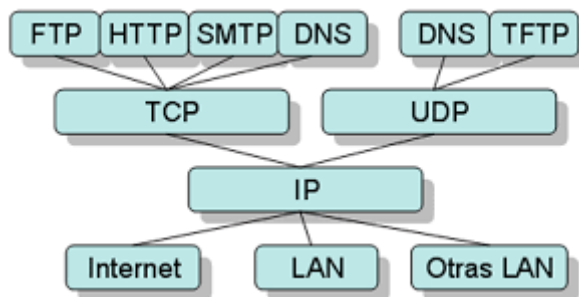


Figura 8. Protocolos de la red

Para finalizar, de los protocolos explicados anteriormente derivan los tipos de redes que permiten la comunicación en nuestros dispositivos.

Mediante [5] Los tipos de redes de comunicaciones de datos se clasifican según su alcance (LAN, MAN, WAN y WLAN)

## LAN (RED DE AREA LOCAL)

Son los equipos conectados dentro de un área geográfica pequeña en la misma organización y con la misma tecnología. Generalmente, vía Ethernet.

## MAN (RED DE AREA METROPOLITANA)

Conexión de diversas LAN que se comunican como parte de la misma red de área local. Permiten su conexión mediante Conmutadores o routers (fibra óptica).

## WAN (RED DE AREA EXTENSA)

Conexión de múltiples LAN a través de grandes distancias geográficas. Permiten su conexión a través de routers.

## WLAN (RED DE AREA LOCAL INALAMBRICA)

Conexión de dos puntos a través de ondas electromagnéticas. Permiten su conexión a través de Satélites y microondas.

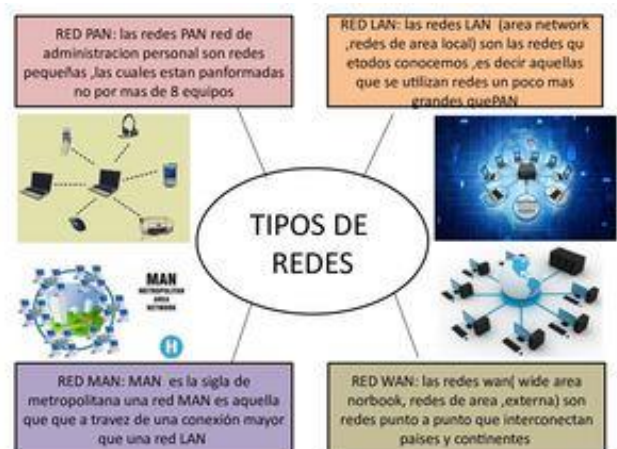


Figura 9. Tipos de redes de comunicación

## I.3 INGENIERÍA DE SOFTWARE

Por definición de [6] La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos, más conocidos como software.

En otras palabras, 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. Podríamos decir que la ingeniería del software es el continente donde se aloja el contenido, que sería el software en sí.

La ingeniería de software cuenta con un ciclo de vida con la finalidad de ordenar el caos que este era anteriormente, para esto cuenta con los procesos de construcción, entrega y evolución del software de manera que el conjunto de estos procesos de como resultado el desarrollo de una aplicación.

Los principales procedimientos según el desarrollo de software basado en [7] son:

- **Definición de objetivos:** definir el resultado del proyecto y su papel en la estrategia global.
- **Análisis de los requisitos y su viabilidad:** recopilar, examinar y formular los requisitos del cliente y examinar cualquier restricción que se pueda aplicar.
- **Diseño general:** requisitos generales de la arquitectura de la aplicación.
- **Diseño en detalle:** definición precisa de cada subconjunto de la aplicación.
- **Programación** (programación e implementación): es la implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.
- **Prueba de unidad:** prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.
- **Integración:** para garantizar que los diferentes módulos se integren con la aplicación. Este es el propósito de la *prueba de integración* que está cuidadosamente documentada.
- **Transición:** Es el momento de la implementación y el desarrollo para los clientes o usuarios. Deben tener tiempo para familiarizarse con el nuevo software.
- **Mantenimiento:** para todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del *software* (mantenimiento continuo).

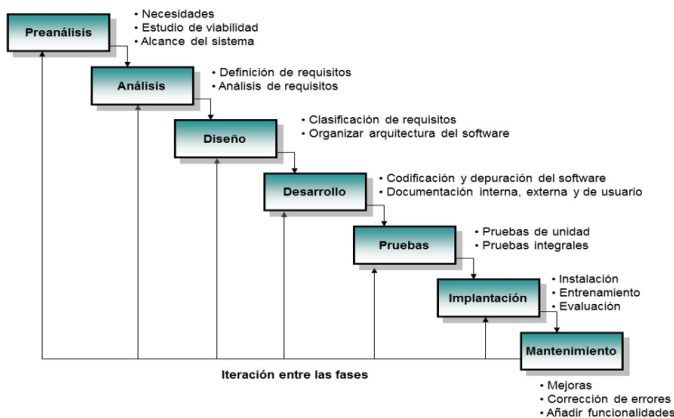


Figura 10. Ciclo de vida del desarrollo de software

Como en todo proceso contamos con diferentes modelos, los cuales a la hora de desarrollar un programa nos ayudaran a un mejor funcionamiento y eficacia, se toma con referencia [8] para explicar más detalladamente los siguientes modelos:

## MODELO DE CASCADA

El primer modelo de proceso de desarrollo de software que fue publicado se deriva de procesos de ingeniería de sistemas más generales (Royce, 1970). Debido a la cascada de una fase a otra, este modelo es conocido como modelo de cascada o como ciclo de la vida del software.

Las ventajas del modelo en cascada son que la documentación se produce en cada fase y que este cuadra con otros modelos del proceso de ingeniería. Su principal problema es su inflexibilidad al dividir el proyecto en distintas etapas. Se deben hacer compromisos en las etapas iniciales, los que hace difícil responder a los cambios en los requerimientos del cliente.



Figura 11. Modelo de cascada

## MODELO DE PROTOTIPO

Los modelos nacieron como un método para acelerar la definición de los requerimientos del software por construir. Un prototipo es un programa. La idea principal es hacer un modelo de la aplicación y presentársela al cliente, sobre todo a nivel de interfaces y otras salidas. El cliente hará sus observaciones sobre lo que se ve en este modelo, y el programa se modificara de acuerdo con dichas observaciones.

El proceso es repetido hasta haber logrado todos los requerimientos del producto en un software, pasando luego a construir la aplicación.

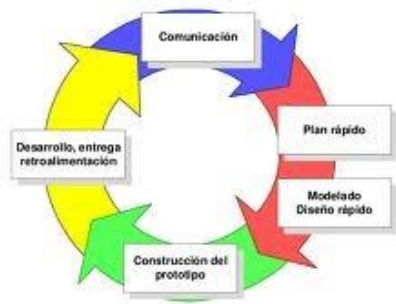


Figura 12. Modelo de prototipo

### MODELO DE ESPIRAL

El modelo espiral busca pasar controladamente, y de manera cíclica, por cuatro actividades en el desarrollo de la aplicación.

El proceso de construcción del software es evolutivo. Empieza con la actividad de la planificación. Después se evaluarán los riesgos existentes. Si el riesgo de continuación del proyecto es muy grande, este se detendrá.

En caso contrario, se sigue con la actividad de ingeniería, que construirá un producto sujeto a la posterior evaluación del cliente. De las observaciones de estos últimos se hará una nueva planificación, se estudiarán los nuevos riesgos, se hará el siguiente producto de ingeniería. Una vez que el cliente de por aprobado el producto, se planificará la puesta en producción, se evaluarán sus riesgos y se concluirá, en la actividad de ingeniería, con la puesta de producción de la aplicación.

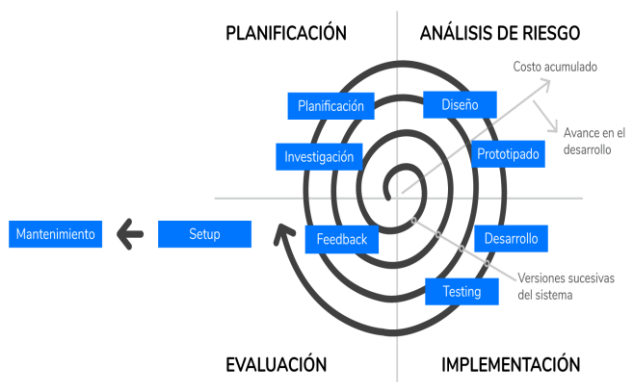


Figura 13. Modelo en espiral

## I.4 INTELIGENCIA ARTIFICIAL

Se define según [9] como la combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano. Una tecnología que todavía nos resulta lejana y misteriosa, pero que desde hace unos años está presente en nuestro día a día a todas horas.

La inteligencia artificial se encuentra clasificada en diferentes tipos que proveen su funcionalidad. Estos tipos son los siguientes:

### SISTEMAS QUE PIENSAN COMO HUMANOS

Automatizan actividades como la toma de decisiones, la resolución de problemas y el aprendizaje. Un ejemplo son las redes neuronales artificiales.

### SISTEMAS QUE ACTUAN COMO HUMANOS

Se trata de computadoras que realizan tareas de forma similar a como lo hacen las personas. Es el caso de los robots.

### SISTEMAS QUE PIENSAN RACIONALMENTE

Intentan emular el pensamiento lógico racional de los humanos, es decir, se investiga cómo lograr que las máquinas puedan percibir, razonar y actuar en consecuencia. Los sistemas expertos se engloban en este grupo.

### SISTEMAS QUE ACTUAN RACIONALMENTE

Idealmente, son aquellos que tratan de imitar de manera racional el comportamiento humano, como los agentes inteligentes.

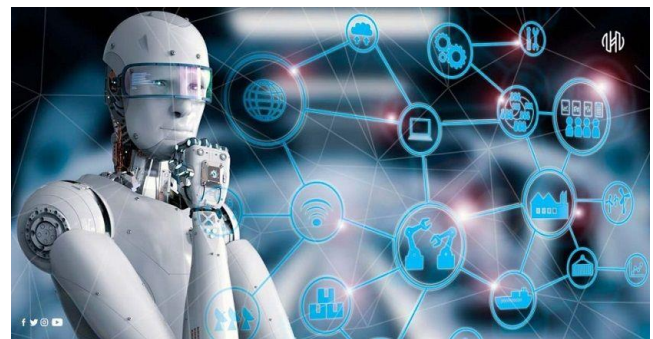


Figura 14. Inteligencia artificial.

### Aplicaciones de la inteligencia artificial

Las técnicas desarrolladas en el campo de la inteligencia artificial son numerosas y ubicuas. Comúnmente cuando un problema es resuelto mediante inteligencia artificial la solución es incorporada en ámbitos de la industria y de la vida diaria de los usuarios de programas de computadora, pero la

percepción popular se olvida de los orígenes de estas tecnologías que dejan de ser percibidas como inteligencia artificial. A este fenómeno se le conoce como el efecto IA.

Publicado por [10] los campos en los que más se emplea la inteligencia artificial son:

- Lingüística computacional.
- Minería de datos (Data Mining).
- Industria.
- Medicina.
- Mundos virtuales.
- Procesamiento de lenguaje natural (Natural Language Processing).
- Robótica.
- Mecatrónica.
- Sistemas de apoyo a la decisión.
- Videojuegos.
- Prototipos informáticos.
- Análisis de sistemas dinámicos.
- Simulación de multitudes.
- Sistemas Operativos.
- Automoción.

[ramaci%C3%B3n%20es,relaciones%20funciones%20instrucciones\).](#)

[3] <https://ed.team/blog/que-son-las-redes-y-como-funciona-internet>

[4] <https://definicion.de/protocolo-de-red/>

[5] <https://www.significados.com/redes/>

[6] <https://definicion.de/ingenieria-de-software/>

[7][https://es.wikipedia.org/wiki/Ingenier%C3%ADa\\_de\\_software#Objetivos](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software#Objetivos)

[8] <https://www.marcoteorico.com/curso/91/ingenieria-de-software/854/modelos-de-la-ingenieria-del-software>

[9]<https://www.iberdrola.com/innovacion/que-es-inteligencia-artificial>

[10][https://es.wikipedia.org/wiki/Inteligencia\\_artificial#Categor%C3%ADas](https://es.wikipedia.org/wiki/Inteligencia_artificial#Categor%C3%ADas)

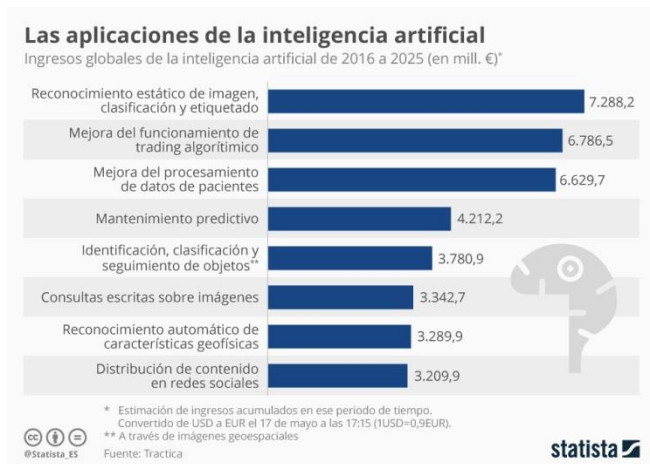


Figura 15. Aplicaciones de la inteligencia artificial.

## 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%20prog>