Sistemas y Computación

Systems and Computing

Autor: Yulian Eduardo Salguero Ramírez

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

Correo-e: yulian.salguero@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.

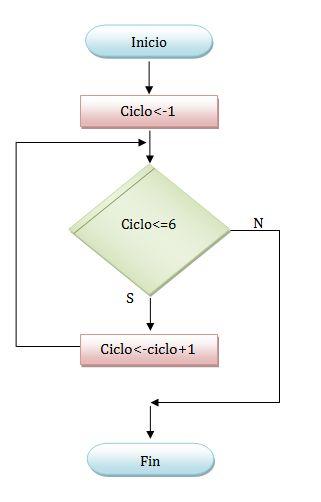


Figura 1. Paradigma estructurado

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.

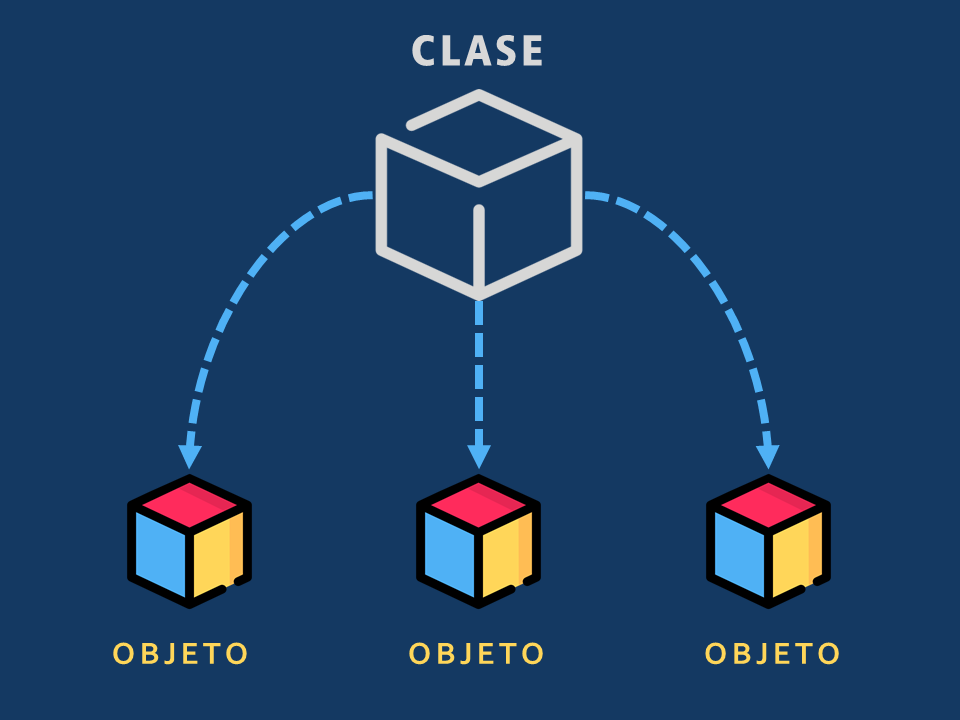


Figura 2. Paradigma orientado a objetos

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.

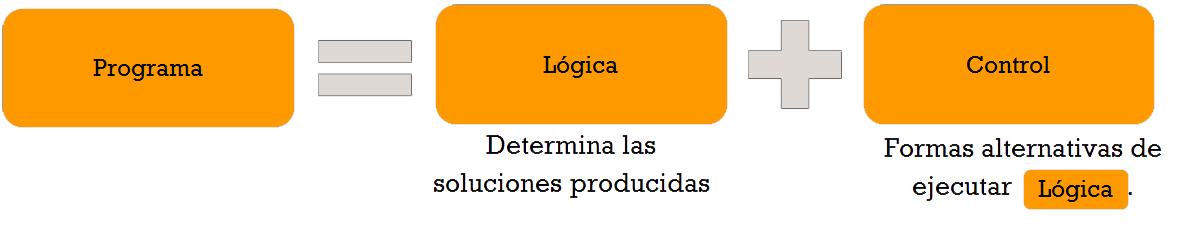


Figura 3. Paradigma lógico

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.

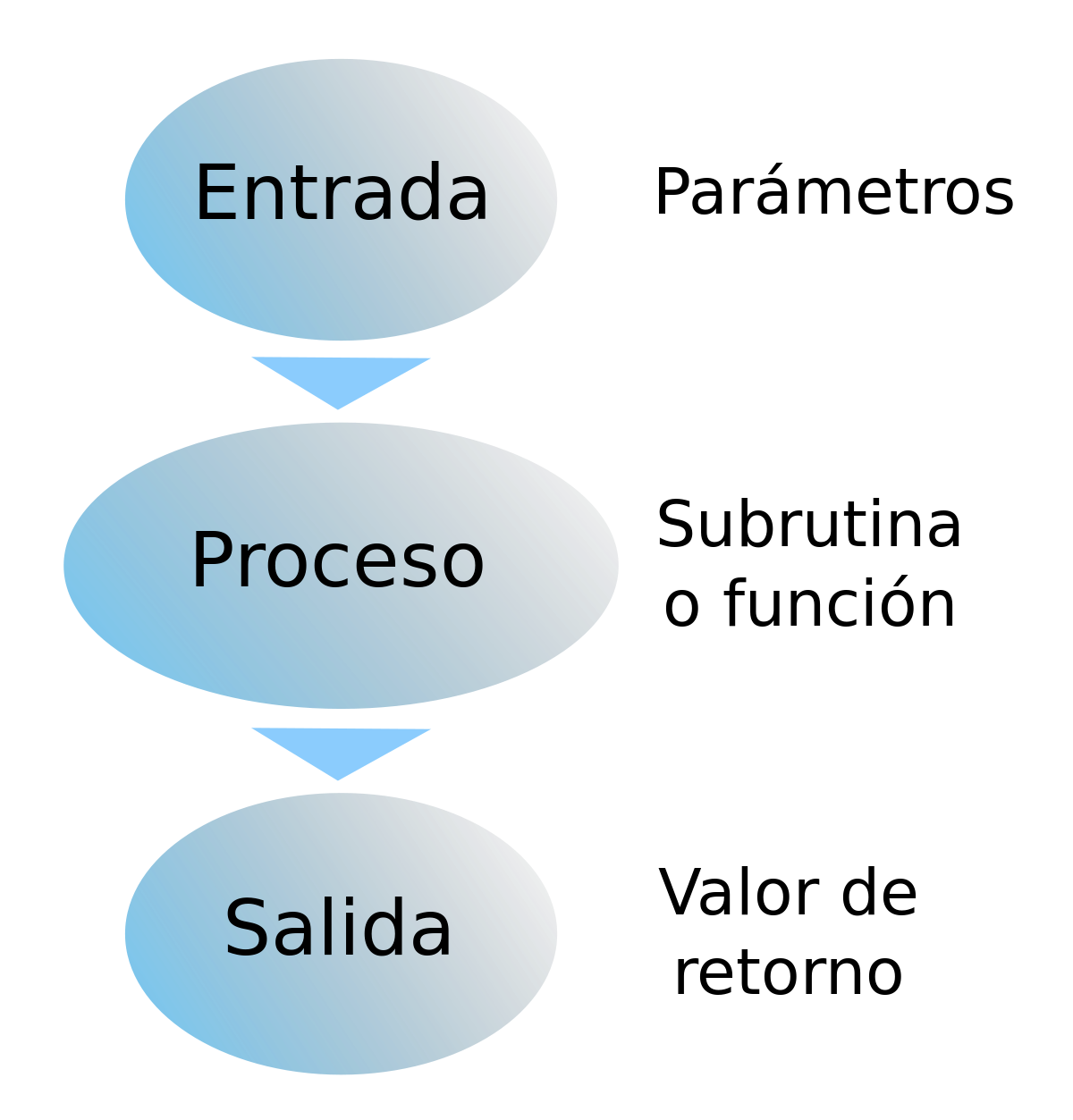


Figura 4. Paradigma funcional.

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 les 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.

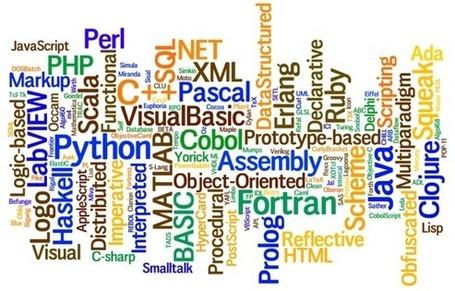


Figura 7. Lenguajes de programación.

* 1. REDES Y COMUNICACIONES

Una red es la combinación de dos o más sistemas y los enlaces de conexión de estos. Una red física es el hardware que configura la red (el equipo, entre el que se incluirían las tarjetas de los adaptadores, los cables y las líneas de teléfono). 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 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.

Las siete capas del modelo OSI se describen del modo siguiente:Diagrama

Descripción generada automáticamente con confianza media

Figura 8. Modelo OSI.

Las redes permiten varias funciones de comunicaciones de aplicaciones y de usuarios, por ejemplo:

* Enviar correo electrónico
* Emular otro terminal o iniciar la sesión en otro sistema
* Transferir datos
* Ejecutar programas que residen en un nodo remoto
* Entrada de datos
* Consultas de datos
* Entrada de proceso por lotes remota
* Compartimiento de recursos
* Compartimiento de datos
* Comunicaciones con otros sistemas operativos
  1. INGENIERÍA DE SOFTWARE

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*. ​Integra ciencias de la computación, ciencias aplicadas y las ciencias básicas en las cuales se encuentra apoyada la ingeniería.

La creación del *software* es un proceso intrínsecamente creativo y la ingeniería del *software* trata de sistematizar este proceso con el fin de acotar el riesgo de fracaso en la consecución del objetivo, por medio de diversas técnicas que se han demostrado adecuadas sobre la base de la experiencia previa.

La ingeniería de *software* se puede considerar como la ingeniería aplicada al *software*, esto es, por medios sistematizados y con herramientas preestablecidas, la aplicación de ellos de la manera más eficiente para la obtención de resultados óptimos; objetivos que siempre busca la ingeniería. No es solo de la resolución de problemas, sino más bien teniendo en cuenta las diferentes soluciones, elegir la más apropiada.

La producción de *software* utiliza criterios y normas de la ingeniería de *software*, lo que permite transformarlo en un producto industrial usando bases de la ingeniería como métodos, técnicas y herramientas para desarrollar un producto innovador regido por metodologías y las buenas prácticas.

La ingeniería de *software* aplica diferentes normas y métodos que permiten obtener mejores resultados, en cuanto al desarrollo y uso del *software*, mediante la aplicación correcta de estos procedimientos se puede llegar a cumplir de manera satisfactoria con los objetivos fundamentales de la ingeniería de *software*.

Entre los objetivos de la ingeniería de *software* están:

* Mejorar el diseño de aplicaciones o *software* de tal modo que se adapten de mejor manera a las necesidades de las organizaciones o finalidades para las cuales fueron creadas.
* Promover mayor calidad al desarrollar aplicaciones complejas.
* Brindar mayor exactitud en los costos de proyectos y tiempo de desarrollo de estos.
* Aumentar la eficiencia de los sistemas al introducir procesos que permitan medir mediante normas específicas, la calidad del *software* desarrollado, buscando siempre la mejor calidad posible según las necesidades y resultados que se quieren generar.
* Una mejor organización de equipos de trabajo, en el área de desarrollo y mantenimiento de *software*.
* Detectar a través de pruebas, posibles mejoras para un mejor funcionamiento del *software* desarrollado.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Figura 9. Creación de software.

* 1. INTELIGENCIA ARTIFICIAL

La **inteligencia artificial** (**IA**) es, en informática, la inteligencia expresada por máquinas, sus procesadores y sus *softwares*, que serían los análogos al cuerpo, el cerebro y la mente, respectivamente, a diferencia de la inteligencia natural demostrada por humanos y ciertos animales con cerebros complejos. 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. Coloquialmente, el término inteligencia artificial se aplica cuando una máquina imita las funciones «cognitivas» que los humanos asocian con otras mentes humanas, como, por ejemplo: «percibir», «razonar», «aprender» y «resolver problemas».Andreas Kaplan y Michael Haenlein definen la inteligencia artificial como «la capacidad de un sistema para interpretar correctamente datos externos, para aprender de dichos datos y emplear esos conocimientos para lograr tareas y metas concretas a través de la adaptación flexible».​ A medida que las máquinas se vuelven cada vez más capaces, tecnología que alguna vez se pensó que requería de inteligencia se elimina de la definición

La inteligencia artificial es una nueva forma de resolver problemas dentro de los cuales se incluyen los sistemas expertos, el manejo y control de robots y los procesadores, que intenta integrar el conocimiento en tales sistemas, en otras palabras, un sistema inteligente capaz de escribir su propio programa. Un sistema experto definido como una estructura de programación capaz de almacenar y utilizar un conocimiento sobre un área determinada que se traduce en su capacidad de aprendizaje.

Stuart J. Russell y Peter Norvig diferencian varios tipos de inteligencia artificial:

* **Sistemas que piensan como humanos.** - Estos sistemas tratan de emular el pensamiento humano; por ejemplo, las redes neuronales artificiales. La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas y aprendizaje.
* **Sistemas que actúan como humanos.** - Estos sistemas tratan de actuar como humanos; es decir, imitan el comportamiento humano; por ejemplo, la robótica (El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor).
* **Sistemas que piensan racionalmente.** - Es decir, con lógica (idealmente), tratan de imitar el pensamiento racional del ser humano; por ejemplo, los sistemas expertos, (el estudio de los cálculos que hacen posible percibir, razonar y actuar).
* **Sistemas que actúan racionalmente.** – Tratan de emular de forma racional el comportamiento humano; por ejemplo, los agentes inteligentes, que está relacionado con conductas inteligentes en artefactos.

Diagrama

Descripción generada automáticamente

Figura 10. Cerebro bajo el control 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%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://es.wikipedia.org/wiki/Ingeniería_de_software>

[5] <https://es.wikipedia.org/wiki/Inteligencia_artificial>