

Sistemas y Computación

Systems and Computing

Autor: Gian Franco Posso Giraldo

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

Correo-e: f.posso@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 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.

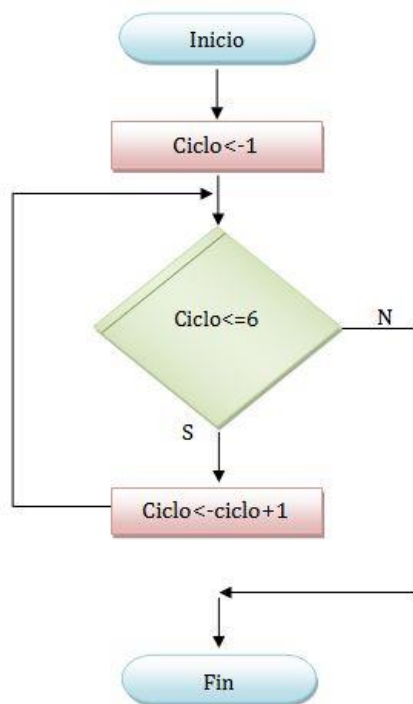


Figura 1. Paradigma estructurado

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.

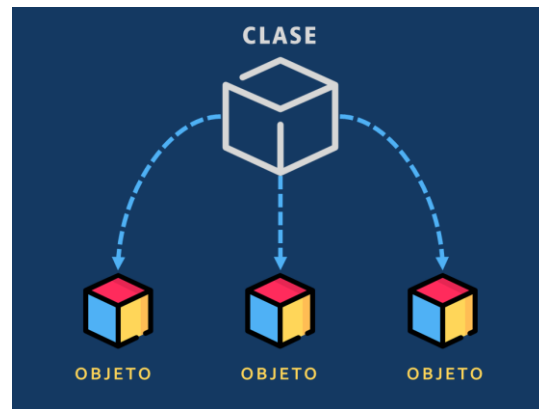


Figura 2. Paradigma orientado a objetos

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.

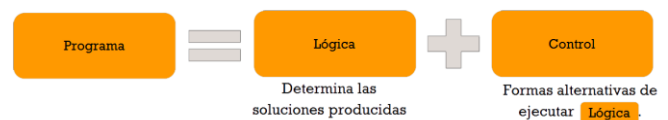


Figura 3. Paradigma lógico

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.

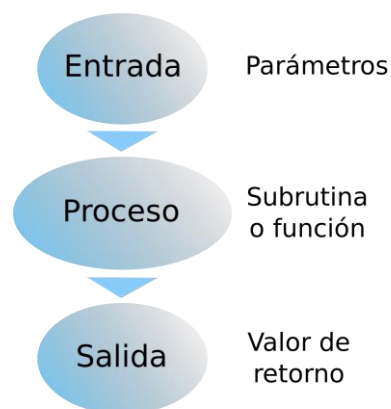


Figura 4. Paradigma funcional.

El paradigma estructurado se conoce, en ciertos entornos, como el paradigma IMPERATIVO. En la siguiente gráfica se aprecia lo visto hasta el momento:

PERSONAL AREA NETWORKS (PAN)

PAN es una red informática para interconectar dispositivos centrados en el espacio de trabajo de una persona individual, pueden utilizarse para la comunicación entre los propios dispositivos personales o para conectarse a una red de nivel superior e Internet donde un dispositivo maestro asume el rol de puerta de enlace. Un PAN puede transportarse a través de buses de computadora cableados como USB.

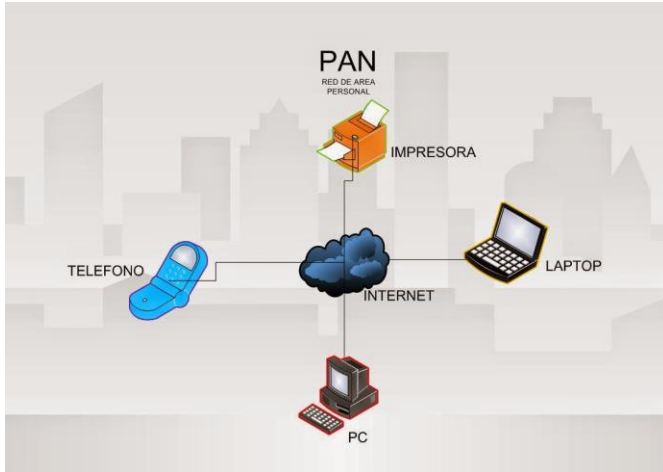


Figura 8. Red tipo PAN

LOCAL AREA NETWORKS (LAN)

LAN es una conexión de dispositivos dentro de un área específica. Cada dispositivo se denomina nodo de la red y está conectado al servidor. Aunque no hay un límite máximo claro para lo que se puede considerar una LAN, esta red cubre típicamente un área pequeña, como una sola oficina, un edificio, o unos pocos edificios dentro de un área.

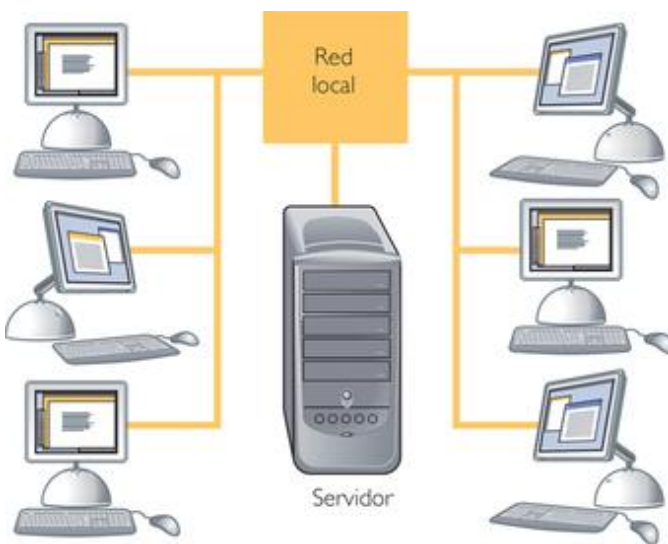


Figura 9. Red tipo LAN

METROPOLITAN AREA NETWORKS (MAN)

MAN es una red de área metropolitana. Este tipo de red es el paso intermedio entre una red LAN y una red WAN, ya que la extensión de este tipo de redes comprende el territorio de una gran ciudad. Las redes MAN son redes de alta velocidad capaces de dar cobertura a una geografía relativamente extensa, aunque nunca superando las dimensiones de una ciudad.

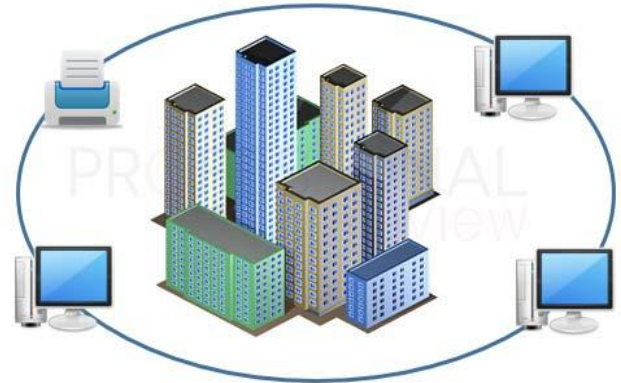


Figura 10. Red tipo MAN

WIDE AREA NETWORKS (WAN)

WAN son redes a gran escala que abarcan países e incluso continentes. No conectan ordenadores individuales, sino otras redes como LAN o MAN. Las WAN pueden ser públicas o estar gestionadas por empresas para conectar varias ubicaciones a grandes distancias.

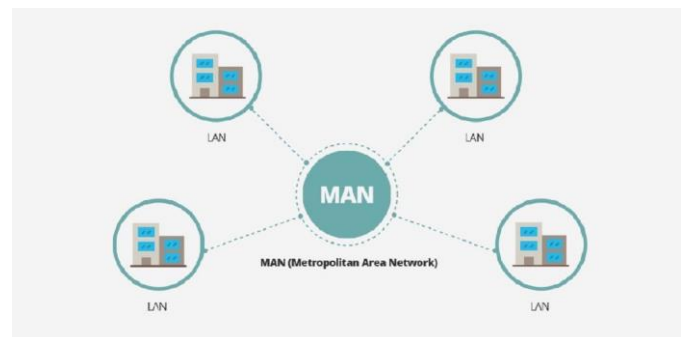


Figura 11. Red tipo WAN

GLOBAL AREA NETWORKS (GAN)

Se trata de la red global de redes, es decir, Internet en sí misma. Ahora bien, no es una conexión individual, pues las empresas con presencia en muchos países también poseen su red de servidores propios.



Figura 12. Red tipo GAN

1.3 INGENIERÍA DE SOFTWARE

[5] La Ingeniería del Software es un área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.

La creación del software es un proceso básicamente 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.

Esta disciplina trasciende la actividad de programación, que es el pilar fundamental a la hora de crear una aplicación. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado.

[6] La ingeniería de software requiere de un ciclo de vida que consiste en la realización de muchas tareas que van en grupos o etapas. Las etapas más comunes para casi todos los modelos de ciclo de vida son las siguientes:

- Análisis de requisitos
- Diseño y arquitectura
- Programación
- Pruebas
- Documentación
- Mantenimiento

ANÁLISIS DE REQUISITOS

Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios. Usualmente el cliente/usuario tiene una visión incompleta/inexacta de lo que necesita y es necesario ayudarlo para obtener la visión completa de los requerimientos. El contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible.

DISEÑO Y ARQUITECTURA

Determinar cómo funcionará de forma general sin entrar en detalles incorporando consideraciones de la implementación tecnológica, como el hardware, la red, etc. Consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la segunda etapa también conocidas como las entidades de negocio.

PROGRAMACIÓN

Se traduce el diseño a código. Es la parte más obvia del trabajo de ingeniería de software y la primera en que se obtienen resultados «tangibles». No necesariamente es la etapa más larga ni la más compleja, aunque una especificación o diseño incompletos/ambiguos pueden exigir que, tareas propias de las etapas anteriores se tengan que realizarse en esta

PRUEBAS

Consiste en comprobar que el software responda/realice correctamente las tareas indicadas en la especificación. Es una buena praxis realizar pruebas a distintos niveles (por ejemplo primero a nivel unitario y después de forma integrada de cada componente) y por equipos diferenciados del de desarrollo (pruebas cruzadas entre los programadores o realizadas por un área de test independiente).

DOCUMENTACIÓN

Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema. Las tareas de esta etapa se inician ya en el primera fase pero sólo finalizan una vez terminadas las pruebas.

MANTENIMIENTO

En esta etapa se realizan un mantenimiento correctivo (resolver errores) y un mantenimiento evolutivo (mejorar la funcionalidad y/o dar respuesta a nuevos requisitos).

Fases del Proceso de Desarrollo del Software

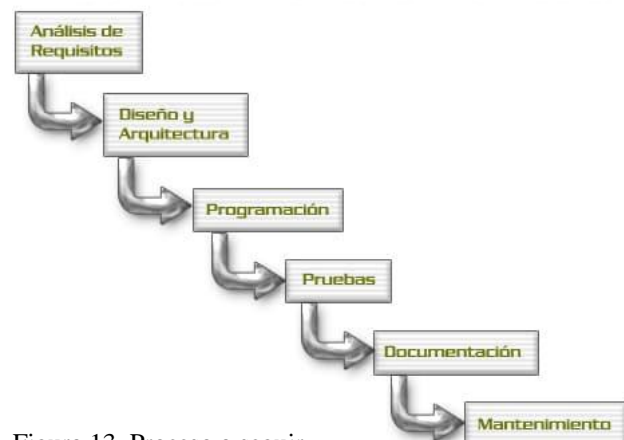


Figura 13. Proceso a seguir.

1.4 INTELIGENCIA ARTIFICIAL

[7] La Inteligencia artificial es el campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes. En otras palabras, la IA es el concepto según el cual “las máquinas piensan como seres humanos”.

Normalmente, un sistema de IA es capaz de analizar datos en grandes cantidades (big data), identificar patrones y tendencias y, por lo tanto, formular predicciones de forma automática, con rapidez y precisión. Para nosotros, lo importante es que la IA permite que nuestras experiencias cotidianas sean más inteligentes. ¿Cómo? Al integrar análisis predictivos y otras técnicas de IA en aplicaciones que utilizamos diariamente.

[8] Aunque no existe una descripción clara y concreta de cómo funciona la inteligencia artificial, sino que depende del tipo de algoritmos y procesos empleados, o de los objetivos perseguidos. En este sentido, se puede hablar de tipos diferentes de inteligencia artificial.

Existen cuatro tipos de tipos de inteligencia artificial. La clasificación propone una visión generalizada sobre los avances en investigación de Inteligencia Artificial. Se trata de una especie de consenso que concluye que las máquinas inteligentes y sensitivas están cada vez más cerca.

Las clasificaciones serían:

- Máquinas reactivas
- Memoria limitada
- Teoría de la mente
- Autoconciencia

MÁQUINAS REACTIVAS

Los tipos más básicos de sistemas de IA son puramente reactivos. No tienen la capacidad de formar recuerdos. Tampoco pueden utilizar experiencias pasadas en las que basar las decisiones actuales.



Figura 14. Deep Blue VS Kasparov

MEMORIA LIMITADA

El Tipo II maneja máquinas que pueden mirar hacia el pasado. Los vehículos autónomos ya hacen algo parecido. Por ejemplo, observan la velocidad y dirección de otros autos. Para que funcionen así hay que identificar objetos específicos y monitorearlos a lo largo del tiempo

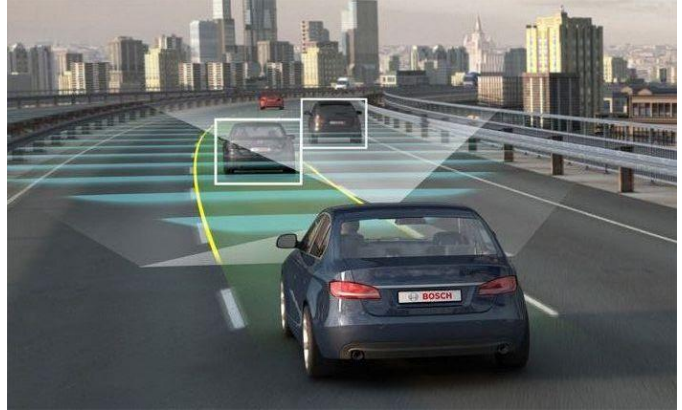


Figura 15. Auto autónomo

TEORÍA DE LA MENTE

Se considera para máquinas que son capaces de comprender que hay personas e individuos con emociones y pensamientos propios. Significando que deberán adaptarse y ajustarse de acuerdo con su entorno



Figura 16. Pepper robot semihumanoide utilizado como recepcionista.

AUTOCONCIENCIA

Se trata de máquinas que tendrán la capacidad de ser conscientes de sí mismas, asumiendo que existen y generando un aprendizaje cognitivo en base a la experiencia propia y de otros individuos. De los tipos de inteligencia artificial, se estima que todavía falta mucho para desarrollar IA de estas características.

Se definieron tres niveles de conciencia humana, basados en el cálculo que ocurre en el cerebro.

- C0, representa cálculos que ocurren sin nuestro conocimiento, como durante el reconocimiento facial,
- C1, implica el llamado conocimiento “global” de la información; en otras palabras, evaluar activamente cantidades de datos para tomar una decisión informada y deliberada en respuesta a circunstancias específicas.
- C2, la autoconciencia surge en este tercer nivel, en el que los individuos reconocen y corrigen los errores e investigan lo desconocido.

En general se considera que cuando podamos explicar, en términos computacionales, cuáles son las diferencias entre los seres humanos conscientes e inconscientes podremos programarlo en las computadoras.

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.monografias.com/trabajos58/redes-comunicaciones/redes-comunicaciones.shtml>
- [4] <https://www.ionos.mx/digitalguide/servidores/known-how/los-tipos-de-redes-mas-conocidos/>
- [5] https://www.ecured.cu/Ingenier%C3%ADa_de_software
- [6] <https://micarrerauniversitaria.com/c-ingenieria/ingenieria-de-software/>
- [7] <https://www.salesforce.com/mx/blog/2017/6/Que-es-la-inteligencia-artificial.html>
- [8] <https://inteligenteyartificial.com/tipos-de-inteligencia-artificial/>