

Sistemas y Computación

Systems and Computing

Autor: Michael Andres Rodriguez Mena

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

Correo-e: m.rodriguez4@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 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.

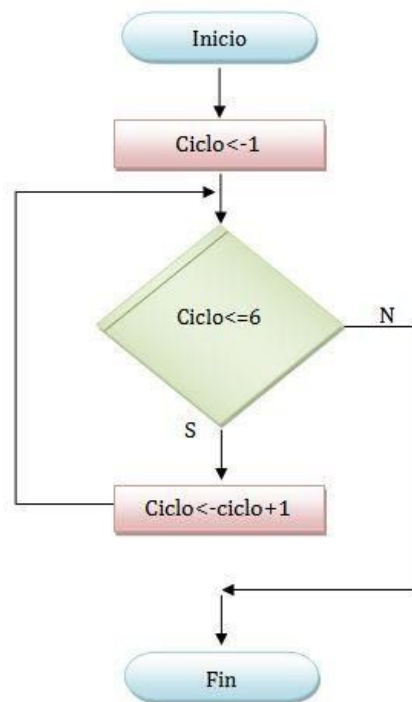


Figura 1. Paradigma estructurado

PARADIGMA DE OBJETOS

El paradigma de objetos es una concepción en la cual 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.

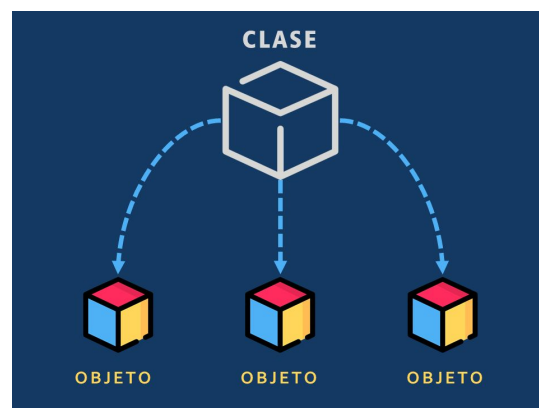


Figura 2. Paradigma orientado a objetos

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.

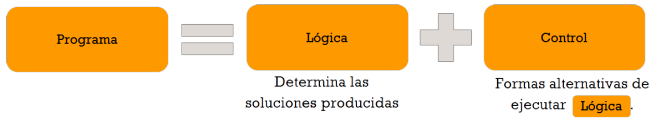


Figura 3. Paradigma lógico

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.

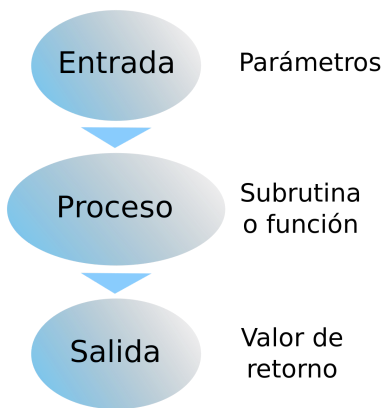


Figura 4. Paradigma funcional.

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.



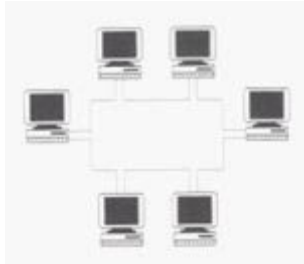
Figura 7. Lenguajes de programación.

1.2 REDES Y COMUNICACIONES

Una red es un conjunto de ordenadores conectados entre sí, que pueden comunicarse compartiendo datos y recursos sin importar la localización física de los distintos dispositivos. A través de una red se pueden ejecutar procesos en otro ordenador o acceder a sus ficheros, enviar mensajes, compartir programas...

Los ordenadores suelen estar conectados entre sí por cables. Pero si la red abarca una región extensa, las conexiones pueden realizarse a través de líneas telefónicas, microondas, líneas de fibra óptica e incluso satélites.

Cada dispositivo activo conectado a la red se denomina *nodo*. Un dispositivo activo es aquel que interviene en la comunicación de forma autónoma, sin estar controlado por otro dispositivo. Por ejemplo, determinadas impresoras son autónomas y pueden dar servicio en una red sin conectarse a un ordenador que las maneje; estas impresoras son nodos de la red.



Dependiendo del territorio que abarca una red se clasifican en:

- LAN: Local Area Network. Está constituida por un conjunto de ordenadores independientes interconectados entre sí, pueden comunicarse y compartir recursos. Abarcan una zona no demasiado grande, un edificio o un campus.
- WAN: Wide Area Network, comprenden regiones más extensas que las LAN e incluso pueden abarcar varios países.

También un conjunto de redes puede conectarse entre sí dando lugar a una red mayor.

Características de una red local

Los ordenadores conectados a una red local pueden ser grandes ordenadores u ordenadores personales, con sus distintos tipos de periféricos. Aunque hay muchos tipos de redes locales entre ellas hay unas características comunes:

1. Un medio de comunicación común a través del cual todos los dispositivos pueden compartir información, programas y equipo, independientemente del lugar físico donde se encuentre el usuario o el dispositivo. Las redes locales están contenidas en una reducida área física: un edificio, un campus, etc.

2. Una velocidad de transmisión muy elevada para que pueda adaptarse a las necesidades de los usuarios y del equipo. El equipo de la red local puede transmitir datos a la velocidad máxima a la que puedan comunicarse las estaciones de la red, suele ser de un Mb por segundo.
3. Una distancia entre estaciones relativamente corta, entre unos metros y varios kilómetros.
4. La posibilidad de utilización de cables de conexión normales.
5. Todos los dispositivos pueden comunicarse con el resto y algunos de ellos pueden funcionar independientemente.
6. Un sistema fiable, con un índice de errores muy bajo. Las redes locales disponen normalmente de su propio sistema de detección y corrección de errores de transmisión.
7. Flexibilidad, el usuario administra y controla su propio sistema.

I.3 INGENIERÍA DE SOFTWARE

Es una especialidad que consiste en sistemas, instrumentos y técnicas que se emplean en el desarrollo de los programas informáticos, también, incorpora el análisis precedente de la situación, el bosquejo del proyecto, el desarrollo del software, el ensayo necesario para comprobar su funcionamiento correcto y poner en funcionamiento el sistema.

Se debe señalar, que el desarrollo del software va unido a lo que se conoce en el campo del software “ciclo de vida del software” que consiste en cuatro etapas que se conocen como: **concepción, elaboración, construcción y transición.**

La **concepción** determina la repercusión del proyecto y diseña el modelo de negocio; la **elaboración** precisa la planificación del proyecto, especificando las características y apoya la arquitectura; la **construcción** es la elaboración del producto; y la **transición** es la entrega del producto terminado a los usuarios.



Al inicio de la informática, el software era un componente más. El programa se consideraba un «arte», para el que no existían métodos, era un procedimiento que se realizaba sin planificación alguna.

Para entonces, todo el programa se desarrollaba con las dimensiones propias para cada necesidad concreta, y por consiguiente, normalmente tenía muy poca difusión, por lo que, solo quien lo necesita, escribía sobre el asunto y este mismo se encargaba de mantenerlo.

Pero ya En la actualidad el software tiene un doble papel. Es el producto, pero al mismo tiempo, actúa como el conductor que entrega el producto. Como conductor utilizado para entregar el producto, actúa como base de control, por ejemplo un sistema operativo, o un sistema gestor de redes.

Diferencia ingeniería de software y sistemas

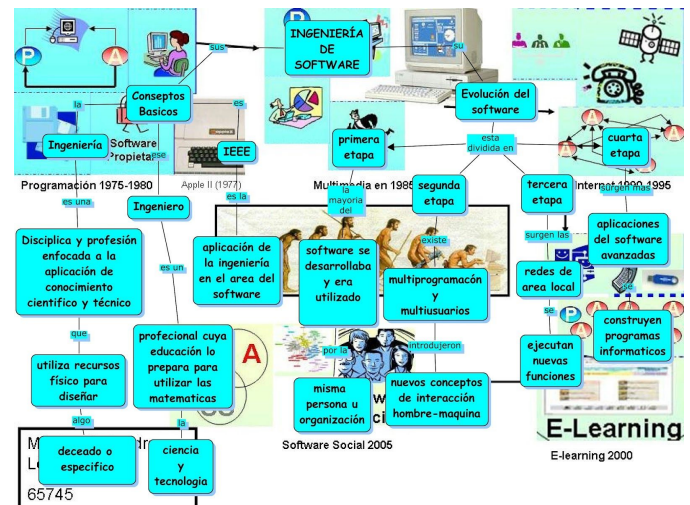
Los ingenieros de sistema están capacitados para ejercer cargos directivos, especialmente en el sector administrativo y también son capaces de ser líderes de una diversidad de proyectos; estudiar y desarrollar programas, asesoría e informática funcional de la organización y así servir de apoyo en las decisiones y realización de procedimientos en la operatividad.



Mientras que la Ingeniería de Software, como ya hemos dicho, es una especialidad que consiste en sistemas, instrumentos y técnicas que se emplean en el desarrollo de los programas informáticos

Ingeniería de software conceptos básicos

La ingeniería de software es una especialidad que consiste en sistemas, instrumentos y técnicas que se emplean en el desarrollo de los programas informáticos.



Conceptos básicos de la Ingeniería de Software.

Esta especialidad manifiesta la actividad del programa, que es la labor fundamental para el momento de la creación de un software.

El ingeniero de software se ocupa de toda la gestión del proyecto para que éste pueda evolucionar en un lapso de tiempo determinado y con los recursos previsto para el proyecto.

La ingeniería de software, también, incorpora el análisis precedente de la situación, el bosquejo del proyecto, el desarrollo del software, el ensayo necesario para comprobar su funcionamiento correcto y poner en funcionamiento el sistema.

Objetivos de la ingeniería de software

Los principales objetivos de la ingeniería de software son:

- Diseñar programas informáticos que se adecúen a las exigencias de la sociedad.



Objetivos de la Ingeniería en Software

- Liderar y acoplar el desarrollo de programaciones complicadas.
- Actuar en todas las fases del ciclo de vida de un producto.
- Computar los costos de un proyecto y evaluar los tiempos de desarrollo.
- Realizar el seguimiento de costes y plazos.
- Liderar equipos de trabajo de desarrollo software.
- Estructurar la elaboración de evidencias que comprueben el perfecto funcionamiento de los programas y que se adaptan a los requerimientos de análisis y diseño.
- Diseñar, construir y administrar bases de datos.

- Liderar y orientar a los programadores durante el desarrollo de aplicaciones.
- Incluir procesos de calidad en los sistemas, calculando métricas e indicadores y chequeando la calidad del software producido.
- Liderar y orientar a los programadores durante el desarrollo de aplicaciones.
- Incluir procesos de calidad en los sistemas, calculando métricas e indicadores y chequeando la calidad del software producido.
- Estructurar e inspeccionar el trabajo de su equipo de los técnicos de mantenimiento y los ingenieros de sistemas y redes.

I.4 INTELIGENCIA ARTIFICIAL

La **Inteligencia Artificial** es la **simulación de inteligencia humana por parte de las máquinas**.

Dicho de otro modo, es la **disciplina que trata de crear sistemas capaces de aprender y razonar como un ser humano**, aprendan de la experiencia, averigüen cómo resolver problemas ante unas condiciones dadas, contrasten información y lleven a cabo tareas lógicas.

El hecho de que un sistema posea hardware humanoide y actúe físicamente como tal es un campo perteneciente a la robótica, y se aleja del concepto de Inteligencia Artificial, que se centra en emular el modo de pensar y razonar de los humanos.

Cabe destacar que, de conseguir que un sistema pudiera aprender y pensar como un humano, poseería notables ventajas sobre éste gracias a su velocidad y capacidad de cálculo

EJEMPLOS Y CASOS DE USO DE INTELIGENCIA ARTIFICIAL

Dispositivos en el hogar

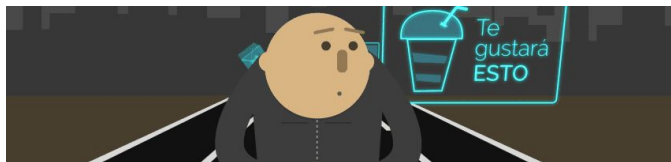
Desde termostatos inteligentes hasta aspiradoras que pasean por todos los rincones de nuestra casa. La **domótica** es uno de los campos “sencillos” de la Inteligencia Artificial que los usuarios hemos podido disfrutar desde hace muchos años.

Filtros de SPAM

No es una de las IA más llamativas, pero empresas como Google o Microsoft aplican multitud de algoritmos en constante evolución con el objetivo de detectar correos fraudulentos y de tipo SPAM.

Anuncios personalizados

Sistemas de IA principalmente impulsados por buscadores y redes sociales, son aquellos que analizan toda la información que disponen del usuario, para mostrarle anuncios con los que probablemente interactuará.



Agentes expertos

Suelen ser sistemas muy entrenados en una actividad intelectual específica, partiendo de los conocimientos de los

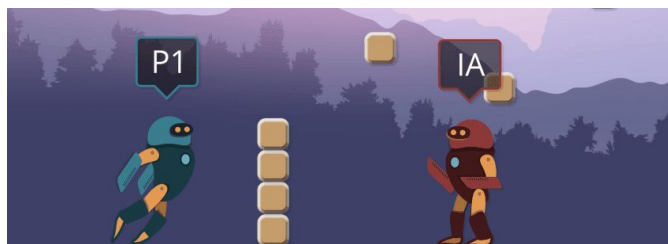
expertos en la materia. Un ejemplo clásico es el de los sistemas que juegan al ajedrez.

Chatbots

Sistemas que realizan un interesante uso del **NLP** (Natural Language Processing) y mejoran con cada experiencia; **permiten una comunicación bidireccional coherente con seres humanos**, ya sea oral o escrita.

Videojuegos

Quizá uno de los más obvios, y que por ello muchas veces queda en el olvido, pero siempre ha sido una de las principales fuentes de mejora de la IA, en esa eterna búsqueda por conseguir que “la máquina” actúe de forma coherente y realista en una partida, ya sea un soldado enemigo o un coche de carreras.

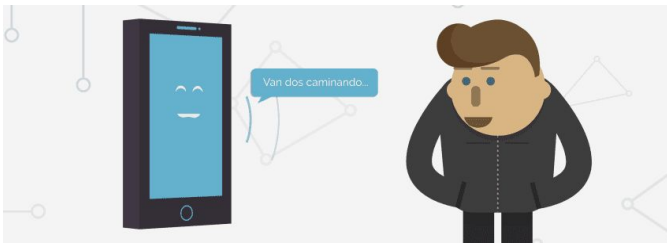


Vehículos autónomos

Son muchas las empresas, y no sólo automovilísticas, que se han subido al carro (nunca mejor dicho) de la automoción inteligente, desarrollando sistemas que procesan ingentes cantidades de datos en tiempo real para determinar una trayectoria correcta del vehículo, prevención de accidentes, etc.

Asistentes virtuales

Es lo más cercano a una IA de película con la que podemos interactuar hoy en día. Reconoce nuestra voz, se adapta a la forma en que pedimos las cosas, y es capaz de recomendarnos entretenimiento acorde a nuestros gustos. Uno de los fuertes de estas tecnologías es que cuentan con **un inmenso número de usuarios** que los alimentan de forma constante y **ayudan a reforzar sus algoritmos de aprendizaje**.



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).)

<https://micarrerauniversitaria.com/c-ingenieria/ingenieria-de-software/>

<https://www.monografias.com/trabajos58/redes-comunicaciones/redes-comunicaciones.shtml>

<https://www.auraportal.com/es/que-es-la-inteligencia-artificial/>