

Juan Noboa

UB96086IN105308

CURSO

Introducción a la Ingeniería en Sistemas

ATLANTIC INTERNATIONAL UNIVERSITY

Honolulu, Hawái

Julio de 2026

Índice

Índice.....	2
Introducción	3
Desarrollo.....	3
1. Fundamentos: la Teoría General de Sistemas	3
2. Computación, algoritmos e ingeniería de software	5
3. Tendencias: inteligencia artificial y panorama de la disciplina.....	7
Actualización y aplicación práctica	8
Conclusión	10
Bibliografía	10

Introducción

Después de más de veinticinco años de ejercer en redes empresariales y sistemas de comunicaciones —como director de TI en instituciones públicas de la República Dominicana y como consultor senior en tecnología—, este curso me ofreció la oportunidad de revisar los fundamentos teóricos de una disciplina que practico a diario, muchas veces de manera intuitiva. La ingeniería en sistemas no es solamente una carrera universitaria: es una forma de pensar que permite abordar problemas complejos identificando los elementos que los componen, las relaciones entre ellos y las propiedades que emergen del conjunto.

El presente ensayo desarrolla los contenidos de las diez lecciones del curso organizados en tres ejes temáticos. El primero aborda los fundamentos conceptuales: qué es un sistema, qué propone la Teoría General de Sistemas (TGS) de Ludwig von Bertalanffy y cómo estas ideas dieron origen a la ingeniería de sistemas como disciplina formal. El segundo eje recorre el terreno de la computación: la resolución de problemas mediante algoritmos, la arquitectura de los sistemas de cómputo y la ingeniería de software como respuesta a la complejidad del desarrollo profesional de aplicaciones. El tercer eje examina las tendencias de la disciplina, en particular la inteligencia artificial y el nuevo papel del ingeniero de sistemas en la sociedad.

A lo largo del documento, y especialmente en la sección de aplicación práctica, contrasto la teoría con mi experiencia profesional en el diseño de redes de datos, con el propósito de demostrar que el enfoque sistémico no es una abstracción académica sino una herramienta de trabajo concreta y vigente.

Desarrollo

1. Fundamentos: la Teoría General de Sistemas

La palabra ingeniería proviene de “ingenio”: la capacidad de aplicar la creatividad para dar solución a una necesidad concreta y fundamentar esa solución. La ingeniería actúa como intermediaria entre la ciencia y la tecnología, aplicando el conocimiento científico al desarrollo de procesos e instrumentos que mejoran el bienestar de la sociedad. Para

comprender la ingeniería en sistemas es necesario partir de cinco conceptos base: la ciencia, como conjunto organizado de conocimientos derivados del método científico; la ingeniería, como su aplicación práctica; el sistema, como objeto de estudio; y la informática y la computación, como los campos donde esta disciplina despliega hoy la mayor parte de su actividad.

Un sistema puede definirse como un conjunto de elementos organizados y relacionados que procesan entradas para producir salidas orientadas a un propósito. Bertalanffy (1968) lo sintetiza como “un conjunto de unidades en interrelación”, mientras que Saussure lo describía ya en 1931 como una totalidad organizada cuyos elementos solo pueden definirse en relación con los demás, según el lugar que ocupan en esa totalidad. Ambas definiciones apuntan a lo mismo: lo esencial de un sistema no son sus partes, sino la organización que las vincula.

La Teoría General de Sistemas, formulada por el biólogo Ludwig von Bertalanffy, es la forma sistemática y científica de aproximación y representación de la realidad desde esa premisa. Se caracteriza por ser holística e integradora: le interesan las relaciones y los conjuntos que emergen de ellas. Sus objetivos son desarrollar una terminología general para describir comportamientos sistémicos, formular leyes aplicables a todos ellos y promover su formalización matemática. La idea no era menor: Bertalanffy aspiraba a que la TGS sirviera de mecanismo de integración entre las ciencias naturales y las sociales, buscando isomorfismos —similitudes estructurales— entre disciplinas distintas. De esa semilla germinaron la cibernética de Norbert Wiener, la teoría de la información de Shannon y Weaver y la dinámica de sistemas de Jay Forrester.

El concepto de sistema, sin embargo, es mucho más antiguo que su teoría. Aristóteles ya sostenía que el todo es más que la suma de las partes; Claude Bernard aportó la noción de homeostasis; James Watt, con su regulador de vapor, materializó la realimentación negativa; y Charles Darwin mostró que los organismos se adaptan continuamente a su ambiente. Checkland (1993) denomina “movimiento de sistemas” a la convergencia de todos estos esfuerzos, cuyo propósito común es enfrentar el problema que el método científico clásico, reduccionista, encuentra tan difícil: la complejidad organizada. En ese marco, Bertalanffy estableció una distinción fundamental

entre sistemas abiertos, que intercambian materia, energía e información con su entorno, y sistemas cerrados, que no lo hacen. Checkland aporta además la distinción entre sistemas duros, de naturaleza técnica y automática, y sistemas suaves, aquellos donde interviene directamente la actividad humana.

En sentido estricto, un sistema es una abstracción: no designa un objeto, sino un modo de considerarlo. Como señala Alexander (1971), la idea de sistema sirve para entender comportamientos que resultan de interacciones entre partes. De ahí se desprenden las características básicas que todo sistema exhibe: un propósito u objetivo; el globalismo o totalidad, por el cual un estímulo en cualquier unidad afecta a todas las demás; la entropía, como desgaste natural por el paso del tiempo; la homeostasis, como capacidad de adaptación al contexto; la sinergia u holismo, que integra las partes en torno a un resultado común; y la jerarquía, la organización y los límites que lo definen frente a su entorno. El emergente sistémico —la propiedad que existe en el todo y en ninguna de sus partes— es quizá la noción más poderosa de todas.

Estas ideas no se quedaron en la biología ni en la filosofía: se convirtieron en herramientas de gestión. Stafford Beer, padre de la cibernética organizacional, definió el sistema viable como aquel capaz de auto-organizarse, autocontrolarse y mantener autonomía frente a un medio cambiante, y desarrolló el Modelo de Sistemas Viables (MVS) inspirado en el cuerpo humano: cinco sistemas que van de la operación básica a la identidad y las decisiones políticas de la organización. El modelo es recursivo: todo sistema viable contiene sistemas viables menores y forma parte de otros mayores. Por su parte, la dinámica de sistemas de Forrester entiende la organización como una red de relaciones causa-efecto cuya unidad básica es el ciclo de realimentación. Pensar sistémicamente, en suma, es aprender a ver el mundo con ojos cibernéticos: no centrarse en las partes sino en las interrelaciones, el entorno y las propiedades que emergen del conjunto.

2. Computación, algoritmos e ingeniería de software

Si el primer eje del curso responde a la pregunta de qué es un sistema, el segundo responde a cómo se construyen soluciones dentro de ellos. El punto de partida es la

resolución de problemas. Un problema es una situación cuya respuesta desconocida debe obtenerse mediante métodos sistemáticos, y su resolución exitosa depende ante todo de su correcta comprensión. El matemático George Polya propuso en los años cuarenta una metodología que sigue vigente: comprender el problema, desarrollar un plan, ejecutarlo y evaluar la solución. Trasladada a la informática, esta secuencia se convierte en análisis del problema, diseño del algoritmo y programación.

El algoritmo es el concepto que articula todo el eje: una sucesión finita de pasos no ambiguos, ejecutables en tiempo finito, cuya razón de ser es resolver problemas. Las estrategias de resolución pueden ser algorítmicas —que exploran las alternativas y garantizan soluciones seguras, al costo de mayor esfuerzo— o heurísticas, que acortan el camino apoyándose en la experiencia previa, sin garantía de éxito. La vida cotidiana está llena de problemas algorítmicos: una receta de cocina es un algoritmo, y el problema general de preparar el plato es su problema algorítmico asociado. Esta aparente simplicidad esconde la razón profunda por la que aprendemos a programar: son innumerables los problemas cuya solución puede expresarse mediante un algoritmo.

Las ciencias de la computación dan soporte material a esos algoritmos. Un sistema de cómputo consta de cuatro partes: hardware, software, datos y personas. Es revelador que el propio término lo diga: forman parte de un sistema. Las computadoras, más allá de su tamaño o propósito, hacen esencialmente cuatro cosas —recibir entradas, procesar información, almacenarla y producir salidas—, es decir, reproducen con exactitud el esquema entrada-proceso-salida que la TGS describe para cualquier sistema. En su interior, toda la información se reduce al código binario: el bit como unidad mínima y el byte como agrupación de ocho bits que representa un carácter. Sobre ese fundamento físico se levanta el software: los lenguajes de programación que permiten comunicarse con la máquina, los sistemas operativos que actúan de intermediarios entre el usuario y el hardware, y el software de aplicación que resuelve tareas concretas para las personas.

Cuando el software deja de ser un ejercicio personal y se convierte en un producto profesional, aparece la ingeniería de software. Pressman la define como la disciplina que ofrece métodos y técnicas para desarrollar y mantener software de calidad, y el estándar

IEEE (1993) la precisa como “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software”. La necesidad de esta ingeniería se explica por cinco factores que distinguen el desarrollo profesional del amateur: el volumen de los proyectos, su evolución constante ante los cambios del entorno, su complejidad, la coordinación de equipos numerosos y la comunicación con un cliente que no siempre sabe expresar lo que necesita. El ciclo de vida de todo sistema de información atraviesa tres fases genéricas: definición (qué hay que desarrollar), construcción (diseño, codificación y prueba) y mantenimiento, que coincide con su vida útil.

3. Tendencias: inteligencia artificial y panorama de la disciplina

El tercer eje del curso mira hacia adelante. La inteligencia artificial se ha abordado históricamente desde dos perspectivas: la ingenieril, que busca que las máquinas realicen tareas para las cuales se supone necesaria la inteligencia humana, y la científica, que pretende simular y comprender los procesos inteligentes del ser humano. Las definiciones clásicas recogidas en el curso ilustran esa dualidad: desde “máquinas con mente” (Haugeland, 1985) hasta “el estudio de los cálculos que permiten percibir, razonar y actuar” (Winston, 1992). El rasgo común de los programas que llamamos inteligentes es el manejo de símbolos agrupados como conceptos e ideas, y no como colecciones de datos carentes de significado: procesamiento simbólico, algoritmos de búsqueda, bases de conocimiento y algoritmos de inferencia y aprendizaje. De allí se desprenden ramas como la robótica, la visión artificial, los sistemas expertos, la simulación y las redes neuronales.

Resulta inevitable, leyendo estas definiciones desde el presente, constatar cuánto ha avanzado el campo. La IA que describe el curso es fundamentalmente simbólica: sistemas expertos que codifican reglas de especialistas humanos. La IA actual, dominada por modelos generativos entrenados sobre volúmenes masivos de datos, ha invertido el paradigma: ya no le damos las reglas a la máquina, sino los datos, y ella induce las reglas. Este ensayo es en sí mismo un testimonio de ese cambio: siguiendo la recomendación de la propia universidad de emplear herramientas de IA en la investigación, utilicé asistentes de inteligencia artificial para explorar los materiales del

curso, lo que confirma la tesis de fondo de la lección: la IA dejó de ser una rama especulativa de la computación para convertirse en infraestructura cotidiana del trabajo intelectual.

El panorama de la disciplina completa el cuadro. La ingeniería de sistemas se formaliza en el siglo XX —el concepto nace en los laboratorios Bell y estalla en la década del sesenta— y su profesional aparece como un agente de cambio alineado con el desarrollo socioeconómico de su país. La reflexión de Denning sobre los paradigmas cambiantes es la que encuentro más vigente: el ingeniero de sistemas ya no es solamente un experto que resuelve problemas de hardware y software para producir sistemas más rápidos, baratos y confiables; es un profesional que se ocupa de las preocupaciones de las personas respecto de la información: su privacidad e integridad, los estándares para comunicarla e intercambiarla, y los valores compartidos por quienes usan los computadores y las redes. La tecnología cambió; la responsabilidad, todavía más.

Actualización y aplicación práctica

Mi trayectoria profesional —más de veinticinco años en redes empresariales, como director de TI del Ministerio de Educación de la República Dominicana, del ISFODOSU y del INDOTEL, y como consultor en diseño de sistemas de comunicaciones— es, vista desde este curso, ingeniería de sistemas aplicada en estado puro. Una red de datos corporativa es un sistema de libro de texto: sus elementos (enrutadores, conmutadores, enlaces, servidores, usuarios) importan menos que sus interrelaciones, y su comportamiento global —disponibilidad, latencia, capacidad— es un emergente sistémico que ninguna pieza posee por sí sola. Un conmutador de última generación no garantiza nada; la organización coherente del conjunto lo es todo.

El marco conceptual del curso describe con precisión fenómenos que enfrento en cada proyecto. El esquema entrada-proceso-salida con realimentación es, literalmente, la operación de una red: el tráfico entra, se procesa y enruta, y sale hacia su destino, mientras los sistemas de monitoreo devuelven información de estado que ajusta el comportamiento —realimentación negativa en el sentido exacto de Wiener. La homeostasis se materializa en los protocolos de enrutamiento dinámico y los

mecanismos de redundancia: cuando un enlace falla, la red se reconfigura sola para mantener sus variables críticas dentro del rango aceptable, que es precisamente la definición de sistema viable de Beer. La entropía también es real: sin mantenimiento, toda red se degrada —configuraciones obsoletas, parches pendientes, documentación desactualizada— y combatirla consume buena parte del presupuesto operativo de cualquier organización.

La distinción de Checkland entre sistemas duros y suaves me resulta especialmente útil para explicar por qué fallan los proyectos tecnológicos. El diseño de la red es el sistema duro: técnico, medible, gobernado por parámetros. Pero la red existe dentro de un sistema suave: la organización humana que la usa, con sus intereses, resistencias y modelos mentales. He visto diseños técnicamente impecables fracasar porque se analizó el sistema duro con rigor y el sistema suave con ligereza: no se consideró cómo trabajaba realmente la gente, qué aplicaciones consideraba críticas o quién se sentía dueño de la información. El error, en términos del curso, fue reduccionista: optimizar las partes técnicas ignorando las interrelaciones con el entorno humano.

Dos casos concretos de mi carrera ilustran el método sistémico en acción. El primero: en el Ministerio de Educación dirigí la migración de más de doscientos circuitos de frame relay a una red MPLS, junto con la implementación de un sistema integrado de telefonía IP para veintidós mil usuarios. Un proyecto de esa escala fracasa si se aborda circuito por circuito; solo es manejable definiendo primero los límites del sistema y su entorno, modelando las entradas (tráfico de voz y datos con patrones en serie y aleatorios), los procesos (conmutación, enrutamiento, priorización) y las salidas comprometidas (niveles de servicio), y diseñando los ciclos de realimentación —monitoreo, alertas, revisiones de capacidad— que mantienen las variables críticas dentro de su rango: un sistema viable en el sentido de Beer. El segundo: en el INDOTEL diseñé e implementé redes de comunicación para zonas rurales, donde el componente suave del sistema era dominante. La validación no consistió en medir paquetes por segundo, sino en interactuar con los usuarios de las comunidades para entender sus desafíos reales; el sistema técnico solo tenía sentido integrado al sistema de actividad humana que debía servir. De modo similar, durante la pandemia de Covid-19 la implementación de escritorios como

servicio para el acceso remoto fue, en términos de la teoría, un acto de homeostasis organizacional: el sistema se adaptó para mantener su función esencial ante un cambio drástico del entorno.

La lección de fondo de estos años es que el valor del profesional no está en conocer los equipos —ese conocimiento envejece con cada generación tecnológica, como comprobé al pasar de frame relay a MPLS y de la telefonía tradicional a las arquitecturas SIP— sino en dominar el enfoque sistémico, que no envejece.

Conclusión

Este curso me deja tres conclusiones. La primera es que la Teoría General de Sistemas es el fundamento intelectual común de campos que suelen estudiarse por separado: la cibernética, la computación, la ingeniería de software y la inteligencia artificial son ramas de un mismo árbol, y reconocer ese tronco común permite transferir soluciones de un campo a otro, que era exactamente el propósito de Bertalanffy al buscar isomorfismos entre disciplinas.

La segunda es que el pensamiento sistémico es la competencia más durable de un profesional de la tecnología. Los equipos, protocolos y plataformas que dominé al inicio de mi carrera están obsoletos; el hábito de analizar problemas identificando elementos, relaciones, límites, realimentación y propiedades emergentes es hoy más valioso que nunca. La teoría no compite con la práctica: la ordena y la explica.

La tercera es una toma de conciencia sobre el papel del ingeniero de sistemas que quiero ejercer. Como plantea Denning, la profesión ya no consiste en producir sistemas más rápidos y baratos, sino en hacerse cargo de las preocupaciones de las personas alrededor de la información. En mi práctica de consultoría, esto significa diseñar redes pensando no solo en el rendimiento técnico sino en la seguridad, la privacidad y la continuidad del trabajo de quienes dependen de ellas. Ver el mundo con ojos cibernéticos, como propone el curso, es también asumir esa responsabilidad.

Bibliografía

- Atlantic International University (s.f.). Introducción a la Ingeniería en Sistemas, Sesiones 1-10 [Material del curso]. <https://cursos.aiu.edu>
- Beer, S. (1959). Cybernetics and Management. English Universities Press.
- Bertalanffy, L. von (1968). Teoría General de los Sistemas. Fondo de Cultura Económica.
- Checkland, P. (1993). Pensamiento de sistemas, práctica de sistemas. Grupo Noriega Editores.
- Forrester, J. (1961). Industrial Dynamics. MIT Press.
- Hall, A. D. (1962). A Methodology for Systems Engineering. Van Nostrand.
- IEEE (1993). IEEE Standards Collection: Software Engineering. Institute of Electrical and Electronics Engineers.
- Polya, G. (1945). How to Solve It. Princeton University Press.
- Pressman, R. (2010). Ingeniería del software: un enfoque práctico (7.^a ed.). McGraw-Hill.
- Wiener, N. (1948). Cybernetics: or Control and Communication in the Animal and the Machine. MIT Press.