

# Sistemas Software y Complejidad

---

- Sistemas
  - Definición de sistema.
  - Sistemas Naturales y Sistemas Artificiales.
  - Sistemas de Computación.
  - Componentes Principales de cada Sistema.
- La complejidad
  - Principios Generales.
  - Estrategias para atacar la complejidad.



# Sistemas Software y Complejidad

---

- Sistemas
  - **Definición de sistema.**
  - Sistemas Naturales y Sistemas Artificiales.
  - Sistemas de Computación.
  - Componentes Principales de cada Sistema.
- La complejidad
  - Principios Generales.
  - Estrategias para atacar la complejidad.



# Definición de Sistema (1)

---

- Término sobreutilizado y del que más se ha abusado
  - Sistemas políticos, educativos, aviónicos, bancarios, etc.
- Primeras definiciones:
  - Un conjunto u ordenación de cosas relacionadas de tal manera que forman una unidad o un todo orgánico.
  - Un conjunto de hechos, principios, reglas, etc... Clasificados y ordenados de tal manera que muestran un plan lógico uniendo las diferentes partes.
  - Un método o plan de clasificación u ordenación.
  - Una forma establecida de hacer algo; un método, un procedimiento...



# Definición de Sistema (2)

---

***“Un Sistema es un conjunto de elementos vinculados entre sí, que se comportan como un todo, tienen una frontera bien definida y están organizados para lograr un fin determinado.”***

- **Conjunto:** existe más de un elemento que interviene en el sistema.
- **Vinculados:** existen intercambios entre los elementos que también están bien definidos.
- **Comportamiento:** cada elemento tiene un comportamiento dado y otro con respecto a la totalidad.
- **Frontera:** cada elemento puede distinguirse de los demás elementos del sistema.



# Definición de Sistema (3)

---

- **Organizados:** los elementos están organizados entre sí mediante mecanismos y protocolos de comunicación; cada elemento “conoce” su función.
- Un sistema puede ser analizado desde dos puntos de vista.
  - Enfoque analítico: análisis de cada elemento por separado.
  - Enfoque sistémico: análisis global del sistema.
- Las vinculaciones ayudan a definir las fronteras entre elementos. Cuando éstas son débiles, las fronteras no están bien definidas (difusas).
- El comportamiento de un elemento no define el comportamiento del sistema.
  - Una neurona no se comporta como el Sistema nervioso.



# Sistemas Software y Complejidad

---

- Sistemas
  - Definición de sistema.
  - **Sistemas Naturales y Sistemas Artificiales.**
  - Sistemas de Computación.
  - Componentes Principales de cada Sistema.
- La complejidad
  - Principios Generales.
  - Estrategias para atacar la complejidad.



# Sistemas Naturales (1)

---

- Son los sistemas que no fueron creados por el hombre.
- Existen en la naturaleza y sirven a sus propios fines.
- Pueden dividirse en:
  - Sistemas físicos
    - **Sistemas estelares:** galaxias, sistemas solares, etc.
    - **Sistemas geológicos:** ríos, cordilleras, etc.
    - **Sistemas moleculares:** organizaciones complejas de átomos.
  - Sistemas vivos
    - **Animales.**
    - **Plantas.**
    - **Raza Humana.**
    - **Jerarquías de organismos vivos individuales:** hierbas, manadas, tribus, colmenas.



# Sistemas Naturales (2)

---

- El propósito de estudiar sistemas naturales es ayudar a ilustrar y entender mejor los sistemas hechos por el hombre.
- Subsistemas críticos:
  - Reproductor: capaz de dar origen a otros sistemas similares en el cual se encuentra.
  - Frontera: que mantiene unidos a los componentes que conforman el sistema.
  - Inyector: que transporta la materia/energía desde el exterior a través de la frontera.
  - Distribuidor: que trae el material del exterior y lo reparte a cada componente.
  - Convertidor: cambia ciertos materiales que ingresan al sistema en formas útiles.
  - Almacenamiento: almacena la materia/energía durante diferentes períodos.
  - Expulsor: transmite la materia/energía hacia el exterior en forma de desechos.
  - Motor: mueve el sistema o sus partes en relación con todo o parte del medio.



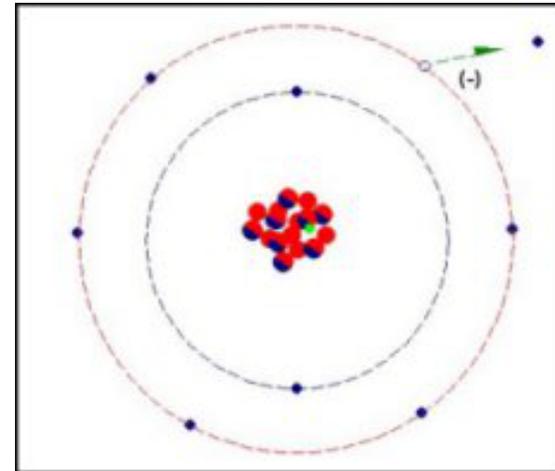
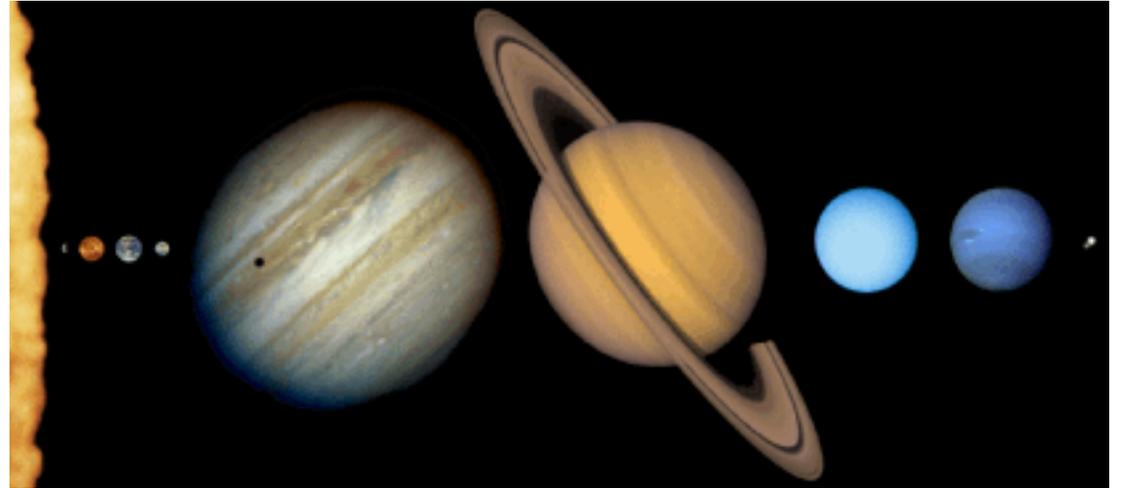
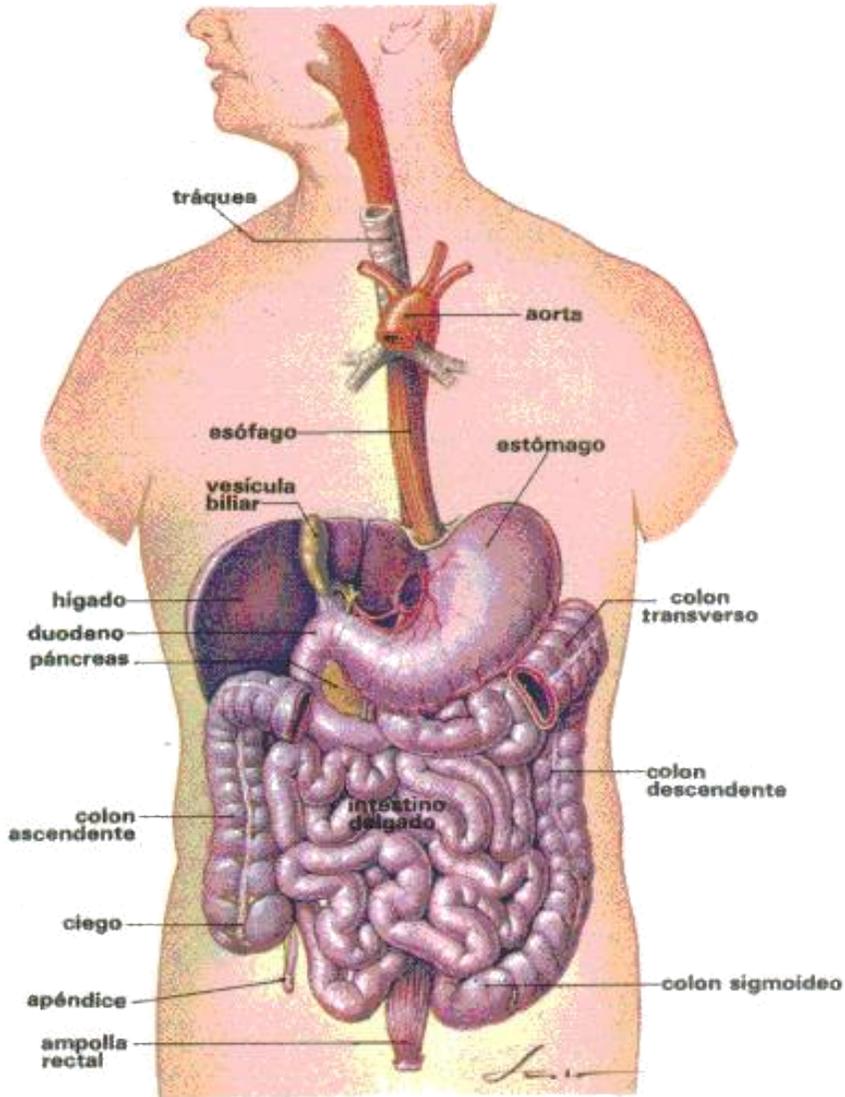
# Sistemas Naturales (3)

---

- Subsistemas críticos:
  - Soporte: mantiene las relaciones espaciales apropiadas entre componentes del sistema.
  - Transductor de entrada: trae las señales portadoras de información al interior.
  - Canal y la Red: compuestos por rutas que transmiten la información al sistema.
  - Asociador: realiza el aprendizaje, formando asociaciones duraderas entre elementos de información dentro del sistema.
  - Memoria: almacena diversos tipos de información.
  - Decisor: el que recibe información de los demás subsistemas y transmite información para controlar así todo el sistema.
  - Transductor de salida: emite señales portadoras de información, desde el sistema hacia el medio exterior.



# Sistemas Naturales (4)



# Sistemas Artificiales (1)

---

- Son los construidos, organizados y mantenidos por humanos; incluyen:
  - Sistemas sociales: organizaciones de leyes, doctrinas, costumbres.
  - Colección organizada y disciplinada de ideas: sistema decimal Dewey.
  - Sistemas de transporte: redes de carreteras, canales, aerolíneas, etc.
  - Sistemas de comunicación: teléfono, telex, señales de humo, etc.
  - Sistemas financieros: contabilidad, inventarios, libro mayor, bolsa de valores.
- Aunque estos sistemas incluyen computadoras, existían antes de que éstas se hubieran inventado.



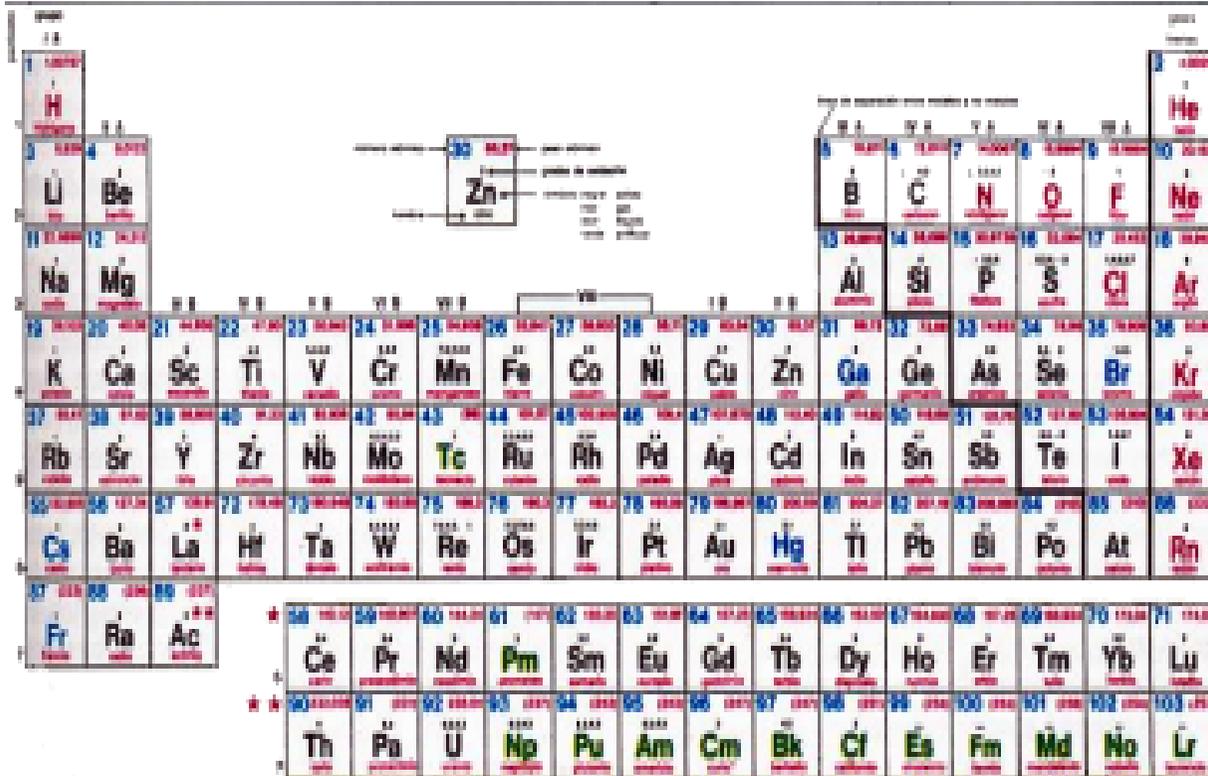
# Sistemas Artificiales (2)

---

- ¿Por qué no deben automatizarse algunos sistemas de procesamiento de información?
  - Costo: puede ser más barato continuar llevando a cabo las funciones en forma manual. No siempre es cierto de que las computadoras sean más económicas.
  - Conveniencia: un sistema automatizado puede ocupar demasiado espacio, hacer demasiado ruido, generar demasiado calor o consumir demasiada energía.
  - Seguridad: si hay datos confidenciales, el sistema tal vez no sea lo suficientemente seguro y sea conveniente guardar la información bajo llave.
  - Facilidad de mantenimiento: no hay personal que pudiera encargarse del mantenimiento (correctivo y preventivo).
  - Políticas: los usuarios podrían pensar que las computadoras amenazan con privarlos de sus empleos.



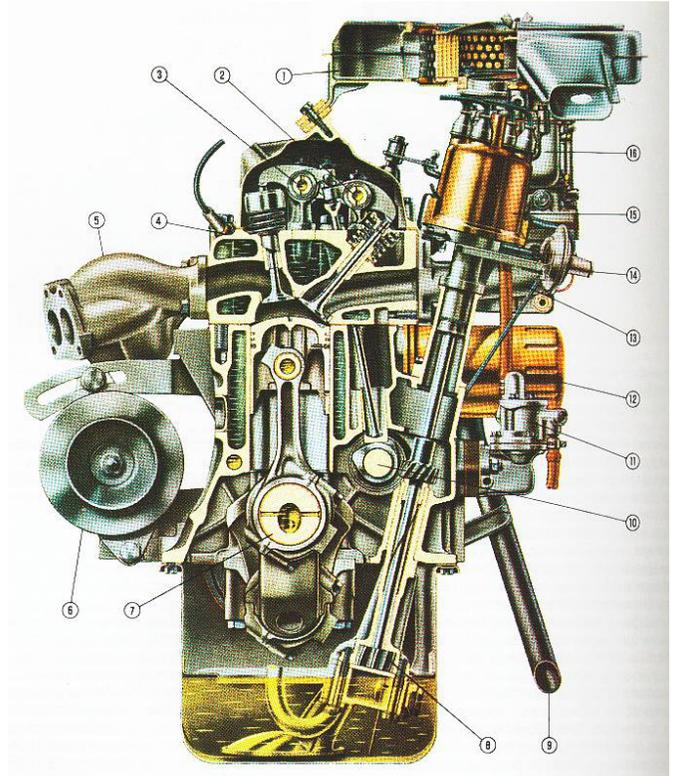
# Sistemas Artificiales (3)



The image shows a standard periodic table of elements. A callout box highlights the element Zinc (Zn) with the following details:

- Symbol: Zn
- Atomic Number: 30
- Atomic Weight: 65.38
- Group: II
- Period: 4
- Block: d
- Classification: Transition metal

1	2											18	19						
H	He											Ar	Kr						
3	4											13	14	15	16	17	18		
Li	Be											B	C	N	O	F	Ne		
11	12											31	32	33	34	35	36		
Na	Mg											Al	Si	P	S	Cl	Ar		
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36		
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr		
37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54		
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe		
55	56	57	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	
Cs	Ba	La	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn		
87	88	89															101	102	103
Fr	Ra	Ac															111	112	113
		58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	
		Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu				
		90	91	92	93	94	95	96	97	98	99	100	101	102	103				
		Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr				



# Sistemas Software y Complejidad

---

- Sistemas
  - Definición de sistema.
  - Sistemas Naturales y Sistemas Artificiales.
  - **Sistemas de Computación.**
  - Componentes Principales de cada Sistema.
- La complejidad
  - Principios Generales.
  - Estrategias para atacar la complejidad.



# Sistemas de Computación (1)

---

- La asignatura se concentra en sistemas automatizados: sistemas hechos por el hombre que interactúan con o son controlados por una o más computadoras.
- Sistemas de Computación:

***“Un conjunto u ordenación de elementos organizados para llevar a cabo algún método, procedimiento o control mediante el procesamiento de información.”***

***“Es la totalidad de los componentes soft/hard que permiten realizar un procesamiento de información.”***



# Sistemas de Computación (2)

---

- Los Sistemas de Computación pueden dividirse u ordenarse por categorías según su aplicación:
  - Sistemas de manufactura, Sistemas de Contabilidad, Sistemas de Defensa, etc.
- Una división en categorías más útil de los sistemas es la siguiente:
  - **Sistemas en línea.**
  - Sistemas de tiempo real.
  - Sistema de apoyo a las decisiones.
  - Sistemas basados en el conocimiento.

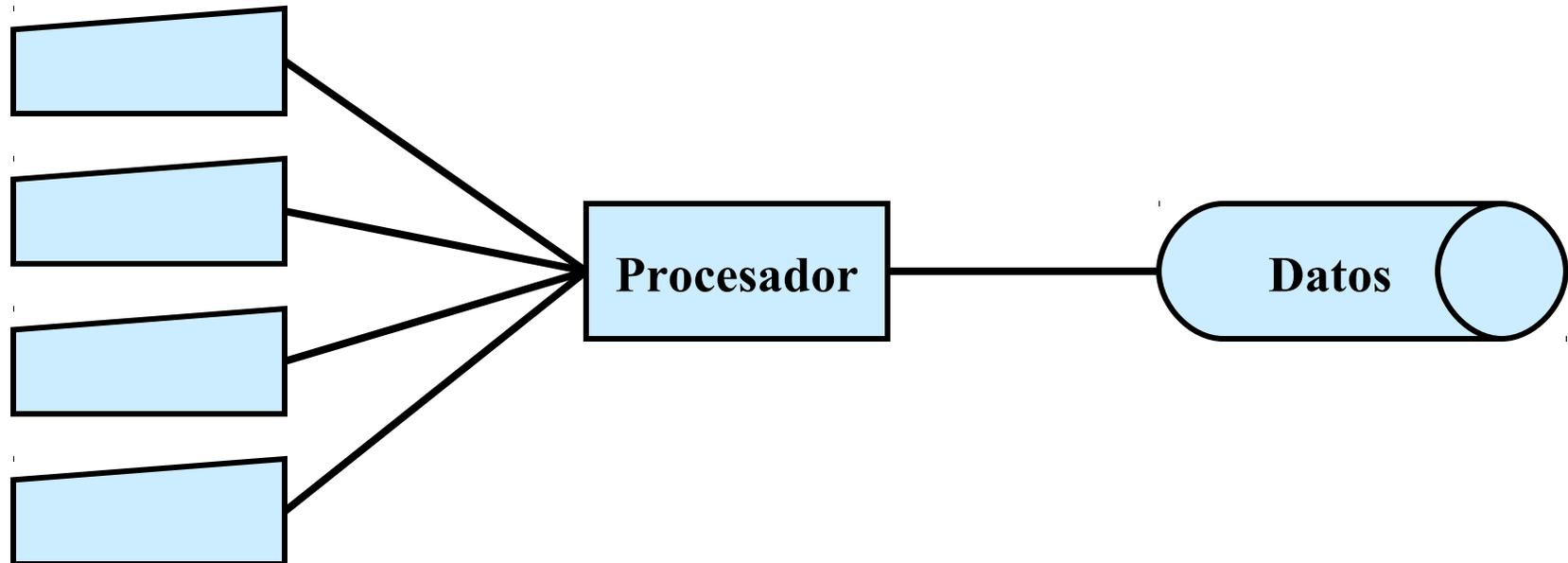


# Sistemas en Línea (1)

---

*“Un sistema en línea es aquel que acepta material de entrada directamente del área donde se creó.”*

*“También es el sistema en el que el material de salida, o el resultado de la computación, se devuelve directamente a donde es requerido.”*



# Sistemas en Línea (2)

---

- Características:
  - Los datos que ingresan o se reciben de la computadora, lo hacen en forma remota.
  - Los datos almacenados se organizan de tal manera que los elementos individuales de información pueden ser recuperados y modificados de forma rápida e individual.
  - Contrastan enormemente con los sistemas fuera de línea o en lotes (batch) (1960-1980) en que el acceso en los sistemas batch es secuencial mientras que en los sistemas en línea es “aleatorio”.
  - Dan suma importancia a la interfaz humano-computadora.
  - Para la rapidez en la recuperación de información, es necesario diseñar las estructuras de datos de la manera más eficientemente posible.



# Sistemas de Computación (2)

---

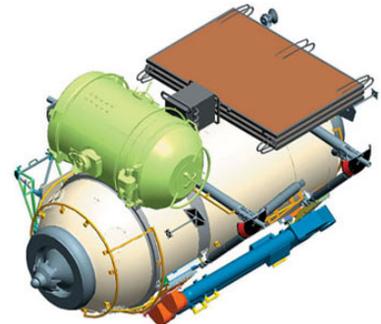
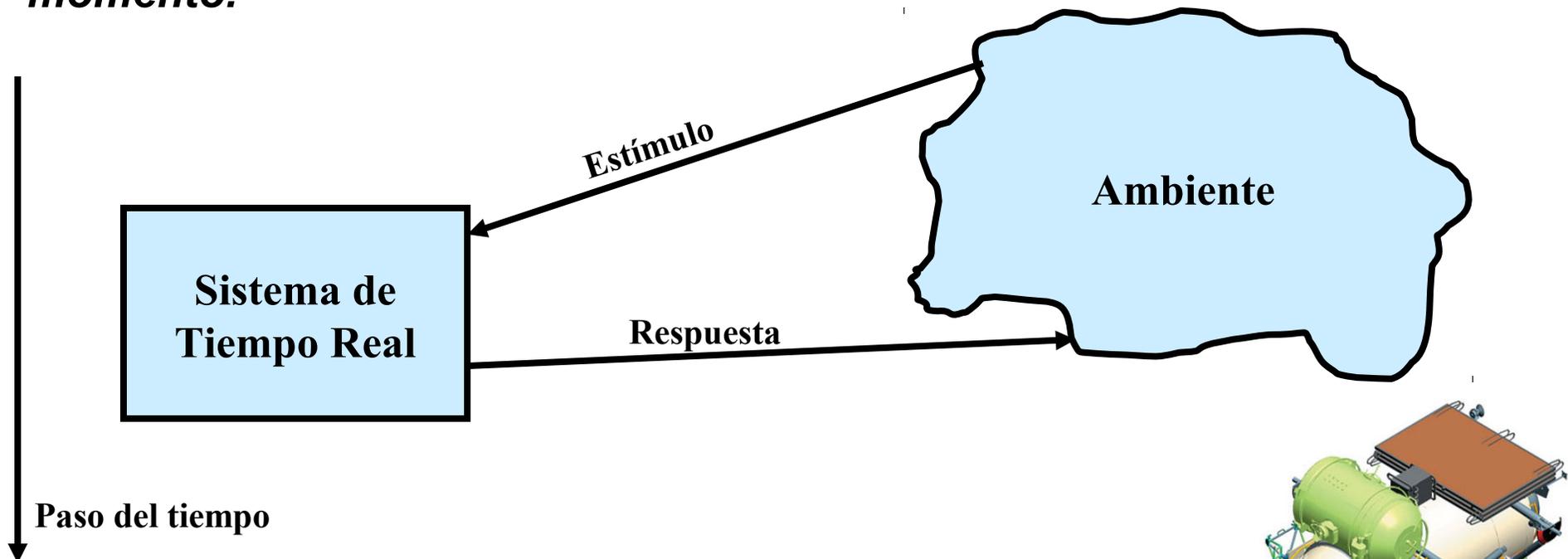
- Los Sistemas de Computación pueden dividirse u ordenarse por categorías según su aplicación:
  - Sistemas de manufactura, Sistemas de Contabilidad, Sistemas de Defensa, etc.
- Una división en categorías más útil de los sistemas es la siguiente:
  - Sistemas en línea.
  - **Sistemas de tiempo real.**
  - Sistema de apoyo a las decisiones.
  - Sistemas basados en el conocimiento.



# Sistemas de Tiempo Real (1)

---

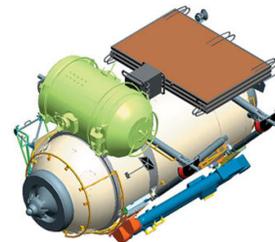
*“Un sistema computacional de tiempo real puede definirse como aquel que controla un ambiente recibiendo datos, procesándolos y devolviéndolos con la suficiente rapidez como para influir en dicho ambiente en ese momento.”*



# Sistemas de Tiempo Real (2)

---

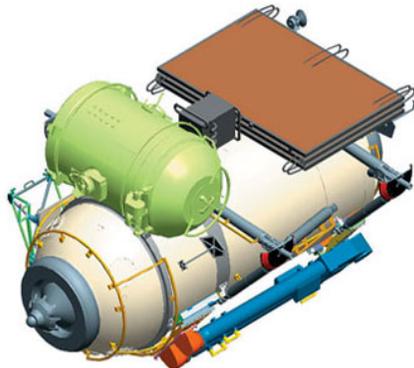
- Diferentes sistemas de tiempo real:
  - Sistemas de Control de Procesos: verificar y controlar refinerías, procesos químicos, molinos y operaciones de maquinado.
  - Sistemas de cajeros automáticos: las “máquinas de efectivo” usadas para hacer depósitos y retiros sencillos en el banco.
  - Sistemas de adquisición de datos de alta velocidad: telemetría de satélites en órbita o captura de datos en experimentos de laboratorio.
  - Sistemas de guía de proyectiles: deben rastrear la trayectoria de un proyectil y hacer ajustes continuos a la orientación y empuje de los propulsores.
  - Sistemas de conmutación telefónica: controlan la transmisión de la voz y datos en miles de llamadas telefónicas.
  - Sistemas de vigilancia de pacientes: detectan signos vitales de pacientes y son capaces de ajustar el medicamento administrado o hacer sonar la alarma.



# Sistemas de Tiempo Real (3)

---

- Características:
  - Si los sistemas no responden rápidamente, el ambiente puede quedar fuera de control.
  - Simultáneamente se lleva a cabo el proceso de muchas actividades.
  - Se asignan prioridades diferentes a diferentes procesos.
  - Se interrumpe una tarea antes de concluirla, para comenzar otra de mayor prioridad.
  - Existe gran comunicación entre tareas (diferentes aspectos de un proceso general).
  - Existe acceso simultáneo a datos comunes, por lo que se requiere sincronización (semáforos).
  - Existe un uso y asignación dinámicos de memoria RAM por cuestiones de economía.



# Sistemas de Computación (2)

---

- Los Sistemas de Computación pueden dividirse u ordenarse por categorías según su aplicación:
  - Sistemas de manufactura, Sistemas de Contabilidad, Sistemas de Defensa, etc.
- Una división en categorías más útil de los sistemas es la siguiente:
  - Sistemas en línea.
  - Sistemas de tiempo real.
  - **Sistema de apoyo a las decisiones.**
  - Sistemas basados en el conocimiento.



# Sistemas de Apoyo a las Decisiones (1)

---

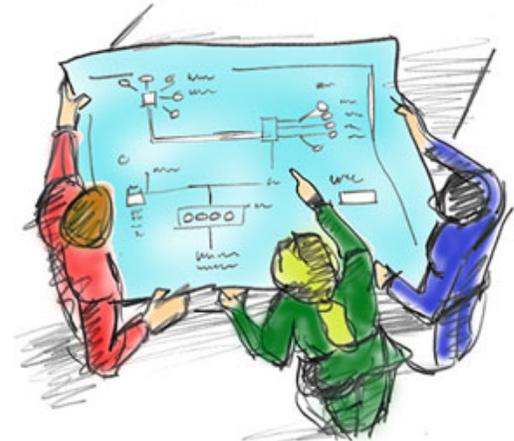
*“Un Sistema Computacional de Apoyo a las Decisiones es aquel que ayuda a los administradores de una organización a tomar decisiones inteligentes y documentadas acerca de los diversos aspectos de la operación.”*

- Algunos sistemas de apoyo a las decisiones:

- Hojas de cálculo
- Análisis Estadístico.
- Pronóstico de Mercados.

- Características

- Siempre proveen información relevante como para que el agente pueda decidir.
- Recuperan y exhiben datos
- Realizan análisis matemáticos y estadísticos de los datos.



# Sistemas de Apoyo a las Decisiones (2)

---

- Características

- Presentan la información en una gran variedad de formatos gráficos.
- Permiten emitir reportes convencionales.
- Permiten articular y mecanizar las reglas utilizadas para llegar a alguna decisión de negocios.
- Se utilizan por los gerentes para evaluar y analizar la misión de la organización.



# Sistemas de Computación (2)

---

- Los Sistemas de Computación pueden dividirse u ordenarse por categorías según su aplicación:
  - Sistemas de manufactura, Sistemas de Contabilidad, Sistemas de Defensa, etc.
- Una división en categorías más útil de los sistemas es la siguiente:
  - Sistemas en línea.
  - Sistemas de tiempo real.
  - Sistema de apoyo a las decisiones.
  - **Sistemas basados en el conocimiento.**



# Sistemas Basados en el Conocimiento

---

*“Un Sistema Computacional basado en el conocimiento tiene como meta producir programas capaces de imitar el desempeño de un ser humano en una gran variedad de tareas inteligentes.”*

*“Los sistemas basados en el conocimiento contienen grandes cantidades de diversos conocimientos que emplean en el desempeño de una tarea dada, imitando el comportamiento de un experto.”*



# Sistemas de Computación (2)

---

- Los Sistemas de Computación pueden dividirse u ordenarse por categorías según su aplicación:
  - Sistemas de manufactura, Sistemas de Contabilidad, Sistemas de Defensa, etc.
- Una división en categorías más útil de los sistemas es la siguiente:
  - Sistemas en línea.
  - Sistemas de tiempo real.
  - Sistema de apoyo a las decisiones.
  - Sistemas basados en el conocimiento.



# Sistemas Software y Complejidad

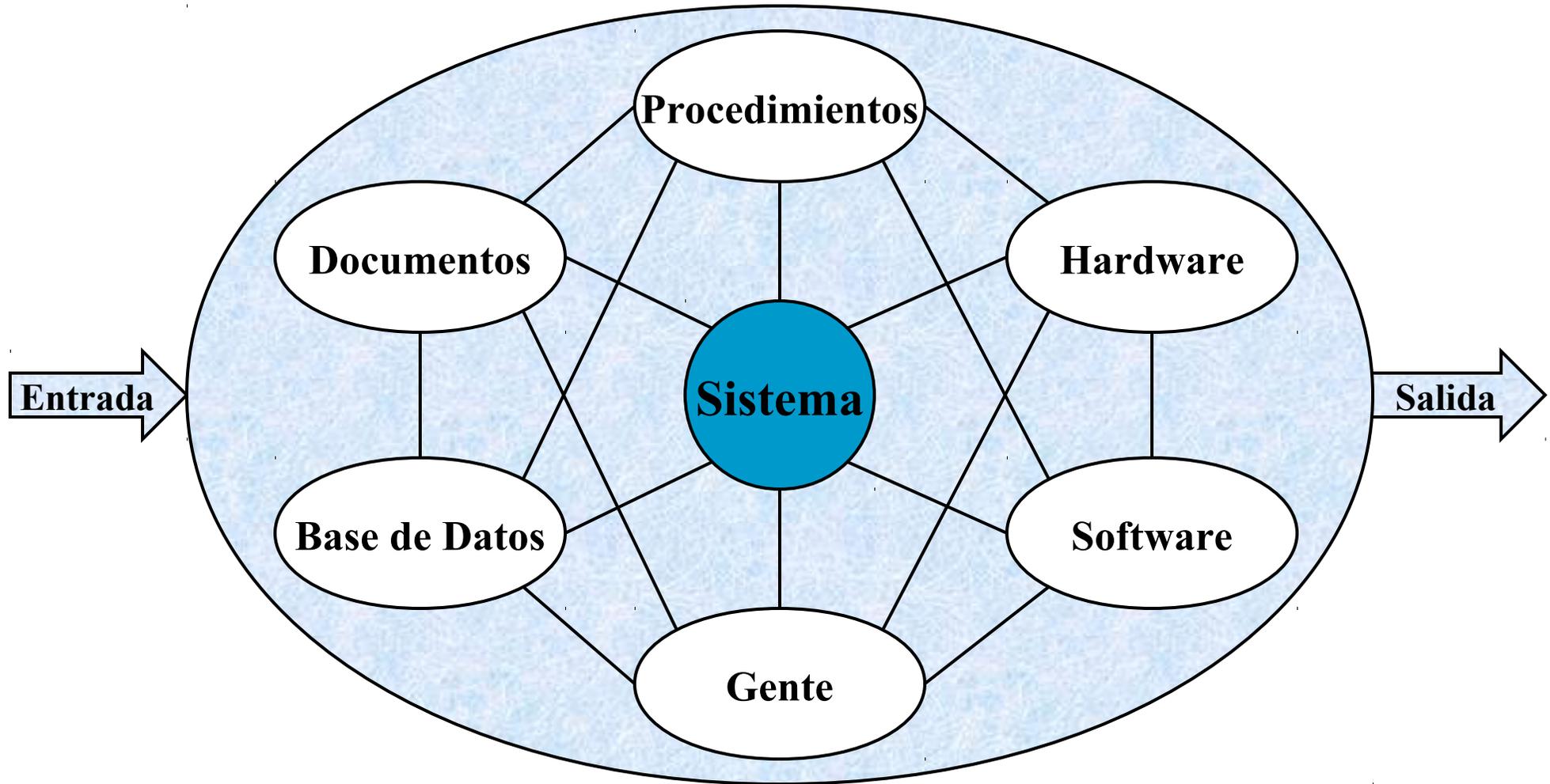
---

- Sistemas
  - Definición de sistema.
  - Sistemas Naturales y Sistemas Artificiales.
  - Sistemas de Computación.
  - **Componentes Principales de cada Sistema.**
- La complejidad
  - Principios Generales.
  - Estrategias para atacar la complejidad.



# Componentes Principales de cada Sistema (1)

---



# Componentes Principales de cada Sistema (2)

---

- **Software:** los programas de computadora, las estructuras de datos y la documentación asociada, que sirven para realizar el método lógico, procedimiento o control requerido.
- **Hardware:** los dispositivos electrónicos (CPU, RAM) que proporcionan la capacidad de computación y los dispositivos electromecánicos (sensores, motores) que proporcionan las funciones del mundo exterior.
- **Gente:** los individuos que son usuarios y operadores del software y del hardware; los expertos en el dominio del problema.



# Gente (1)

---

## ■ Usuarios

- El participante más importante en el juego de los sistemas; es aquel para quien se construye el sistema; es la persona que se tendrá que entrevistar para educir los requisitos.
- Dentro de una organización también se lo conoce como el cliente o “dueño”. Como en muchas otras profesiones, “el cliente siempre tiene la razón.”
- Para identificar al usuario (o usuarios): es aquel que formalmente solicita un sistema.
- El problema principal es que lo que el usuario quiere que el sistema haga puede no serle comunicado al analista, y lo que éste crea que está construyendo para el usuario pudiera no serle comunicado tampoco.
- Hay que evitar suponer que todos los usuarios son iguales ya que, por naturaleza, existe la heterogeneidad de los usuarios (diferentes personalidades, preparación, intereses)
- Es conveniente la clasificación de los usuarios por categoría de trabajo y/o por nivel de experiencia en el procesamiento de datos.



# Gente (2)

---

- Administración

- Administradores usuarios: están a cargo de varias personas en el área operacional donde se va a implantar el nuevo sistema. Desean informes internos y análisis a corto plazo.
- Administradores de informática: encargados del proyecto en sí de sistemas y administradores de nivel superior de la organización de desarrollo de sistemas.
- Administración general: pueden ser el presidente de la organización o el jefe de administración financiera. Se interesan por la planeación estratégica y se concentran más en la información externa. Son los que pagan el desarrollo.
- La interacción entre el analista y los administradores tiene que ver con los recursos que se asignarán al proyecto: personas, tiempo y dinero.
- Cuanto más alto nivel ocupen, menos probable es que sepan (o les interese saber) de la tecnología de computadoras.



# Componentes Principales de cada Sistema (2)

---

- **Software:** los programas de computadora, las estructuras de datos y la documentación asociada, que sirven para realizar el método lógico, procedimiento o control requerido.
- **Hardware:** los dispositivos electrónicos (CPU, RAM) que proporcionan la capacidad de computación y los dispositivos electromecánicos (sensores, motores) que proporcionan las funciones del mundo exterior.
- **Gente:** los individuos que son usuarios y operadores del software y del hardware; los expertos en el dominio del problema.



# Componentes Principales de cada Sistema (3)

---

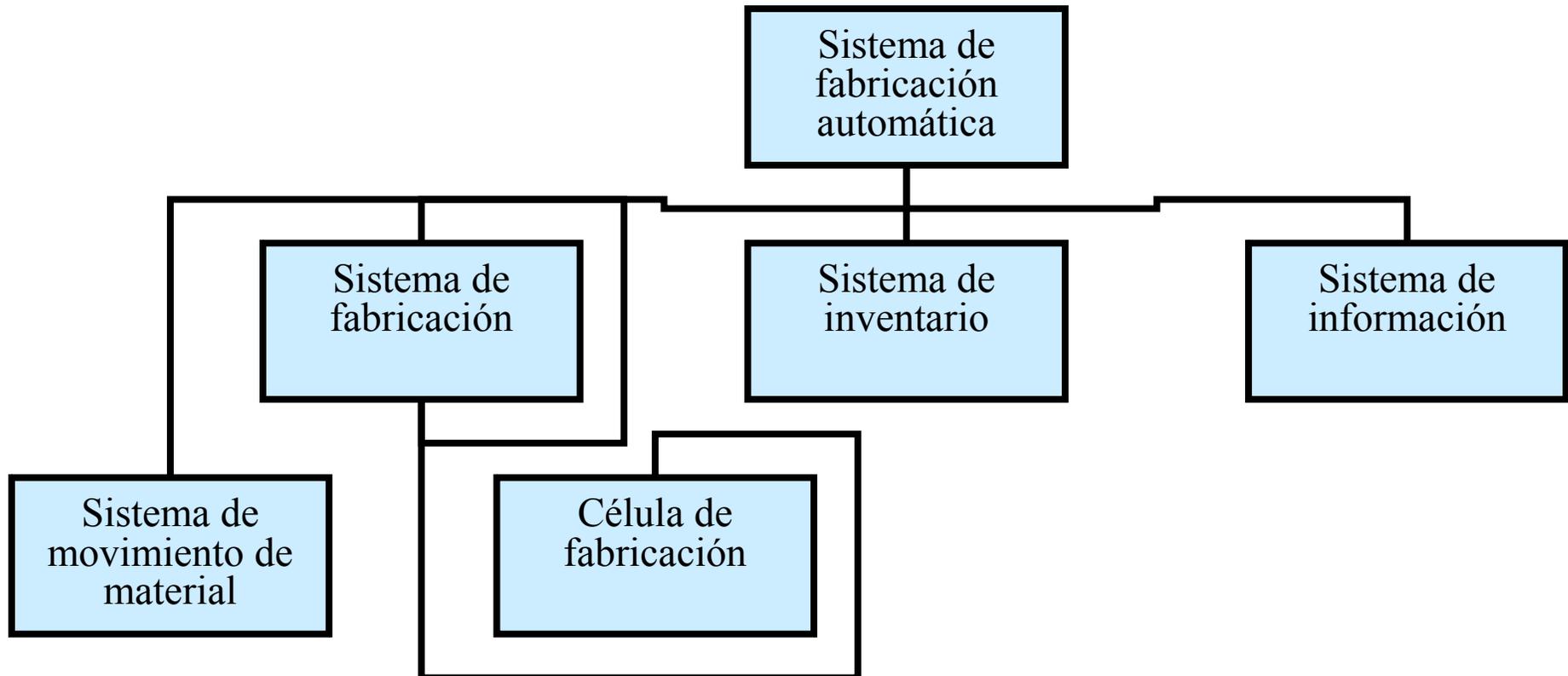
- **Bases de datos:** Una colección grande y organizada de información a la que se accede mediante el software y que es parte integral del funcionamiento del sistema. La información que el sistema recuerda durante un período.
- **Documentación:** los manuales, los impresos, las especificaciones y cualquier otra información descriptiva que explica el uso y/o operación del sistema. *“Lo que no está documentado, no existe.”*
- **Procedimientos:** los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema. Las políticas formales e instrucciones de operación del sistema.



# Componentes Principales de cada Sistema (4)

---

- Los elementos se combinan de muchas maneras para transformar la información.
- Los elementos que componen un sistema pueden también representar un macroelemento de un sistema todavía mayor.



# Componentes Principales de cada Sistema (5)

---

- Un macroelemento es un sistema basado en computadora que forma parte de un sistema basado en computadora.
- El papel del Ingeniero de Sistemas es el de definir los elementos de un sistema basado en computadora específico dentro del contexto de toda la jerarquía de sistemas.



# Sistemas Software y Complejidad

---

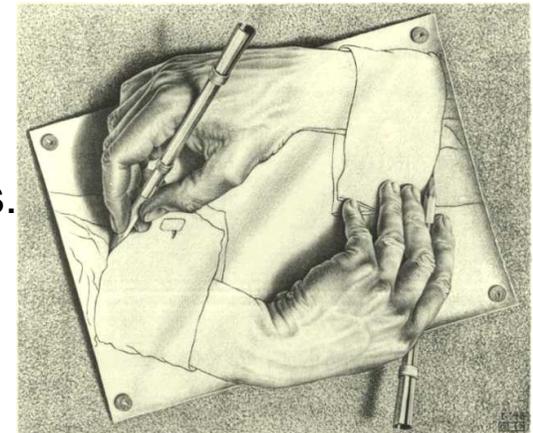
- Sistemas
  - Definición de sistema.
  - Sistemas Naturales y Sistemas Artificiales.
  - Sistemas de Computación.
  - Componentes Principales de cada Sistema.
- La complejidad
  - **Principios Generales.**
  - Estrategias para atacar la complejidad.



# Complejidad. Principios Generales (1)

---

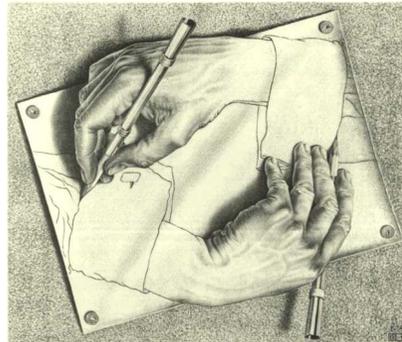
- El software es complejo de forma innata (propia, natural, congénita, no adquirida); la complejidad del software es una propiedad esencial, no accidental.
- La complejidad de los sistemas excede frecuentemente la capacidad intelectual humana.
- La complejidad toma a menudo formas de una jerarquía y ésta, y otras propiedades, nos ayudan a atacar la complejidad.
- Cuanto más complejo sea un sistema, más abierto está al derrumbamiento total. Si no dominamos la complejidad, podemos llegar a:
  - Proyectos retrasados.
  - Exceso en el presupuesto.
  - Productos deficientes respecto a los requerimientos fijados.
  - Desperdicio de recursos humanos.



# Complejidad. Principios Generales (2)

---

- Los cuatro elementos de complejidad del software son:
  1. La complejidad del dominio del problema:
    - Los problemas que se intentan resolver presentan una cantidad muy grande de requisitos que compiten entre sí e incluso se contradicen
    - Existen problemas cuya funcionalidad incluso es difícil de comprender como es el caso del sistema operativo de un avión multimotor o un robot autónomo.
    - La complejidad surge de las grandes dificultades que tienen los usuarios de poder expresar con precisión sus necesidades y los desarrolladores de no conocer el dominio del problema.
    - Una complicación adicional es que los requisitos de un sistema software cambian frecuentemente durante su desarrollo (volatilidad de requisitos) y es preciso seguir su evolución.

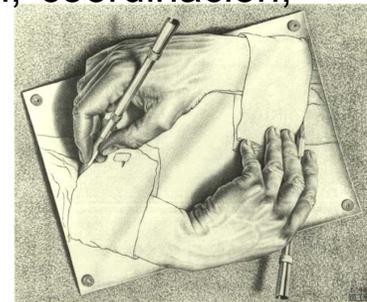


# Complejidad. Principios Generales (3)

---

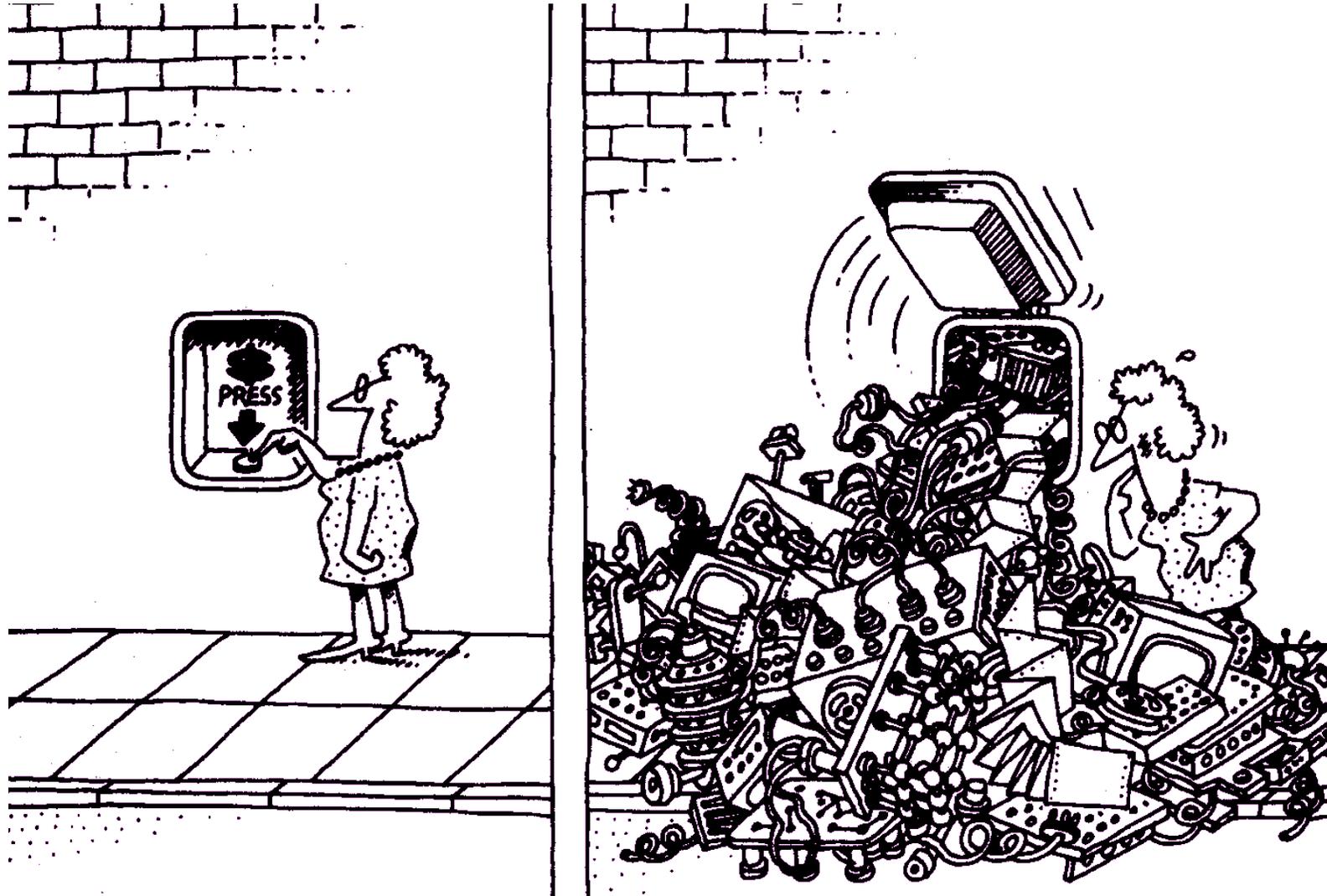
## 1. La dificultad de gestionar el proceso de desarrollo:

- Es tarea del equipo de desarrollo de software ofrecer al usuario la ilusión de simplicidad, esto es, permitir realizar una función al usuario de manera simple ocultando los mecanismos reales complejos que conlleva la realización de dicha función.
- El tamaño no es una gran virtud para un sistema de software. Se hace lo posible por escribir menos código mediante la invención de mecanismos ingeniosos y potentes que dan esta ilusión de simplicidad.
- Sin embargo, a veces es imposible eludir el inmenso volumen de los requerimientos de un sistema y se plantea la obligación de o bien escribir una enorme cantidad de nuevo código o bien reusar código existente.
- Esta cantidad de trabajo exige la utilización de un equipo de desarrolladores con los consecuentes retos asociados al desarrollo en equipo: comunicación, coordinación, ubicación geográfica, dirección, etc.



# Complejidad. Principios Generales (4)

La tarea del equipo de desarrollo de software es ofrecer ilusión de simplicidad.

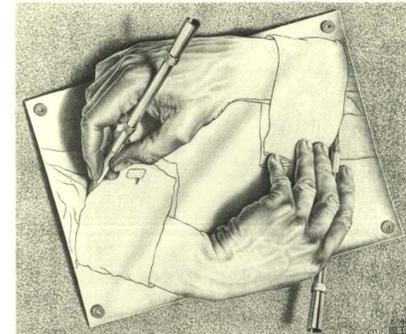


# Complejidad. Principios Generales (5)

---

## 1. La flexibilidad que se puede alcanzar a través del software:

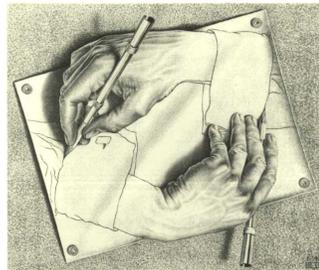
- El software ofrece la flexibilidad máxima por lo que un desarrollo puede expresar casi cualquier clase de abstracción.
- Esta propiedad seduce increíblemente porque empuja al desarrollador a construir todos los bloques fundamentales sobre los que se apoyan estas abstracciones.
- Por ejemplo, realizando una comparación, supongamos que una compañía de construcción de edificios gestiona su propia explotación forestal para cosechar árboles y obtener madera y construye una acería en la obra a fin de hacer vigas a medida para el edificio. Esto suena absurdo, sin embargo, en la industria del software, este comportamiento es frecuente.
- El problema de la flexibilidad reside en la escasez de estándares en la industria del software, y como consecuencia, el desarrollo del software sigue siendo un negocio enormemente laborioso.



# Complejidad. Principios Generales (6)

---

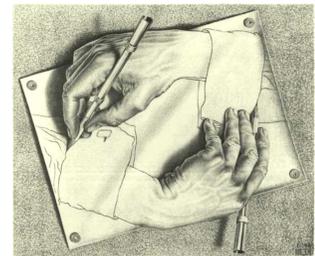
1. Los problemas de caracterizar el comportamiento de sistemas discretos:
  - Al ejecutarse el software en computadoras digitales, se tiene un sistema con estados discretos (miles de variables, así como más de un posible flujo de control)
  - Un sistema discreto tiene un número finito de estados posibles que, en sistemas complejos, puede resultar enorme (explosión combinatoria).
  - Se intenta diseñar el sistema de manera que el comportamiento de una parte del mismo tenga mínimo impacto en el comportamiento de la otra parte, como en los sistemas continuos, donde se puede predecir de manera fiable las salidas (comportamiento) en función de las entradas .
  - Las transiciones entre estados discretos no pueden modelarse como funciones continuas y todos los eventos externos de un software pueden llevar al sistema a un nuevo estado no determinado (Ejemplo del avión)
  - Es imposible probar los sistemas a fondo, debiendo contentarnos con un cierto grado de confianza .



# Complejidad. Principios Generales (7)

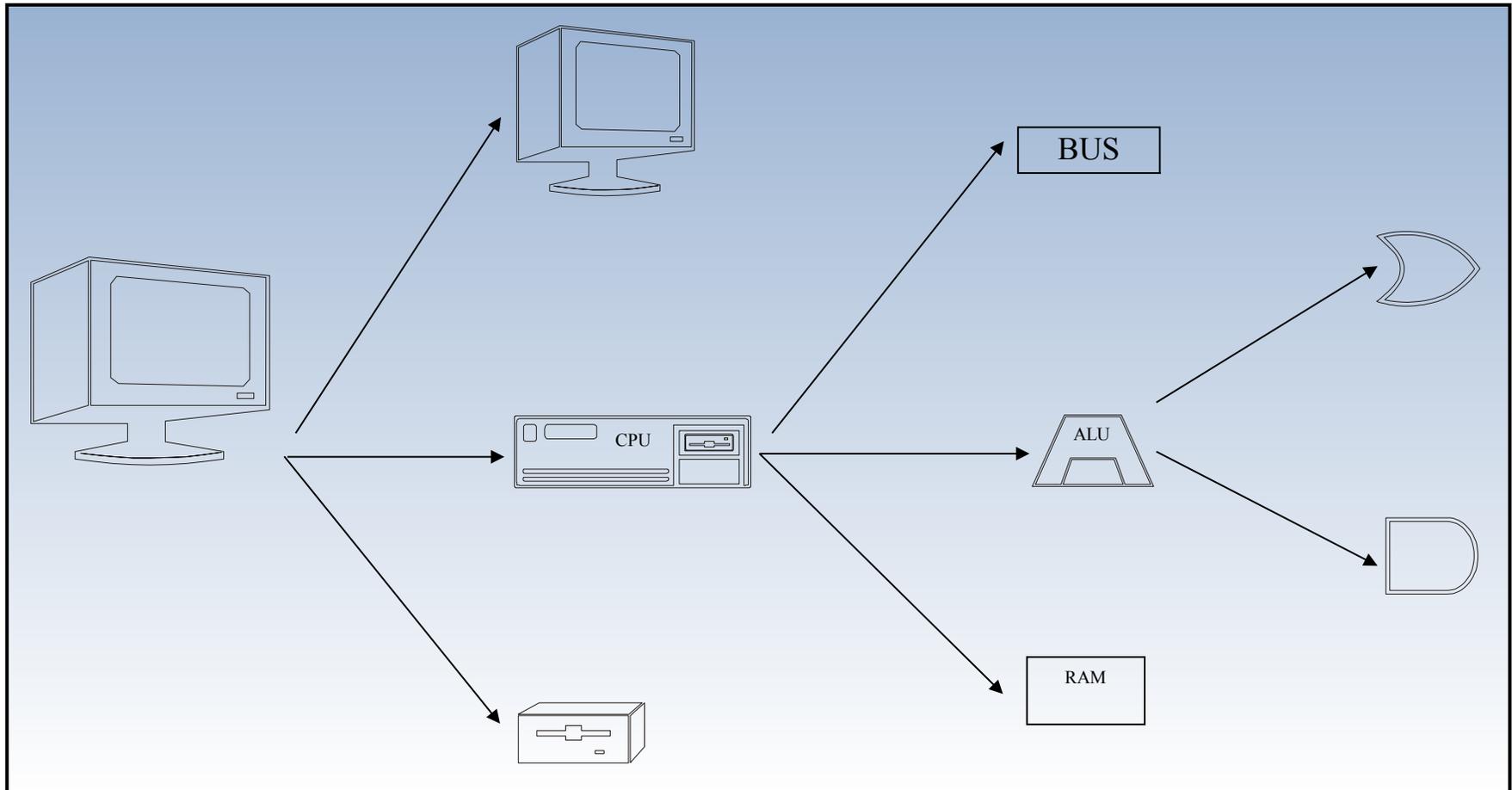
---

- Estructura de los sistemas complejos
  - Estructura de una PC
    - Elementos básicos que a su vez se descomponen en otros elementos.
    - Se aprecia que las compuertas NAND se usan para el diseño de las diferentes partes (CPU, disco rígido, etc.)
    - Naturaleza jerárquica de elementos para formar un sistema. Funciona correctamente sólo merced a la actividad colaboradora de cada una de sus partes principales.
    - Los sistemas complejos no sólo son jerárquicos, sino que los niveles de esta jerarquía representan diferentes niveles de abstracción, cada uno de los cuales se construye sobre el otro y es comprensible por sí mismo.
    - A cada nivel de abstracción, se encuentra una serie de dispositivos que colaboran para proporcionar servicios a capas más altas.



# Complejidad. Principios Generales (8)

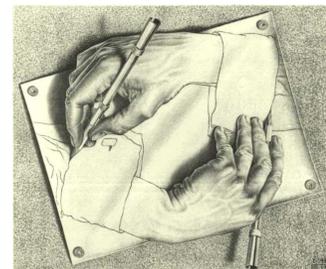
- Estructura de una PC.



# Complejidad. Principios Generales (9)

---

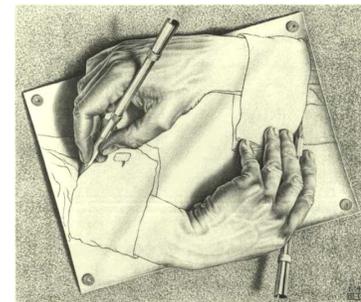
- Estructura de plantas y animales
  - Las plantas están formadas por tres estructuras principales (raíces, tallos y hojas) y cada una de ellas tiene su propia estructura.
  - Las raíces constan de raíces principales, pelos radicales, el ápice y la cofia. La hoja tiene epidermis, mesófilo y tejido vascular.
  - Cada una de estas estructuras se componen, a su vez, de una serie de células, y dentro de cada célula se encuentra otro nivel de complejidad, que abarca elementos como cloroplastos, núcleo, etc.
  - Al igual que la PC, las partes de una planta forman una jerarquía, y cada nivel de esta jerarquía conlleva su propia complejidad.
  - Las partes de un mismo nivel de abstracción interactúan de formas perfectamente definidas (las raíces absorben agua y minerales, los envían a los tallos que luego transportan la materia prima a las hojas) y siempre hay fronteras claras entre el exterior y el interior de determinado nivel de abstracción.



# Complejidad. Principios Generales (10)

---

- En las plantas, las células son los bloques básicos de construcción en todas las jerarquías.
- Todos los elementos primitivos de la planta son células, pero hay muchos tipos de células diferentes: células sin cloroplastos, células con paredes impermeables al agua, etc.
- En una planta no hay partes centralizadas que coordinan las actividades de las partes de nivel inferior.
- Cada parte separada actúa como agente independiente, cada uno de los cuales exhibe un comportamiento bastante complejo, y cada uno de los cuales contribuye a muchas funciones de nivel superior.
- Con la cooperación mutua de varios de estos agentes, se ve la funcionalidad de nivel superior de una planta.
- **Comportamiento emergente:** el comportamiento del todo es mayor que la suma de sus partes.



# Sistemas Software y Complejidad

---

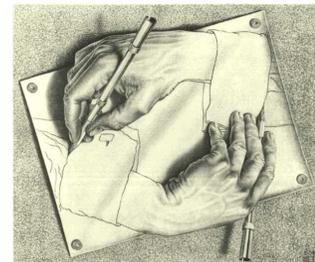
- Sistemas
  - Definición de sistema.
  - Sistemas Naturales y Sistemas Artificiales.
  - Sistemas de Computación.
  - Componentes Principales de cada Sistema.
- La complejidad
  - Principios Generales.
  - **Estrategias para atacar la complejidad.**



# Estrategias para atacar la Complejidad (1)

---

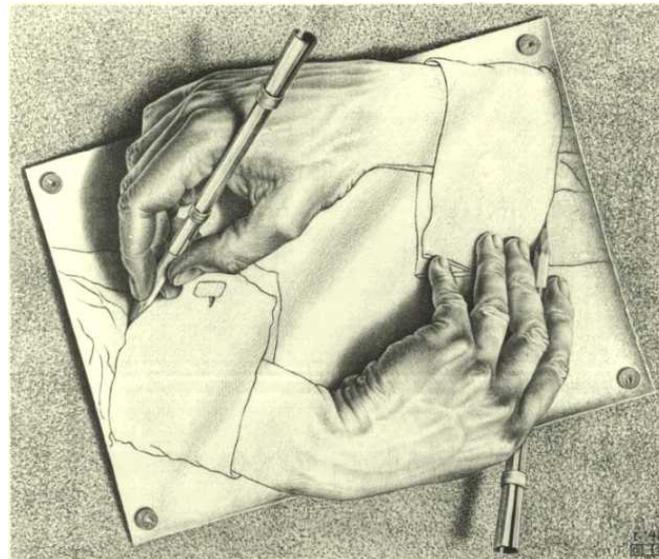
- Los Cinco Atributos que debe tener un Sistema Complejo:
  1. Frecuentemente, la complejidad toma la forma de una jerarquía, por lo cual un sistema complejo se compone de subsistemas relacionados que tienen a su vez propios subsistemas, y así sucesivamente, hasta que se alcanza algún nivel ínfimo de componentes elementales.
  2. La elección de qué componentes de un sistema son primitivos es relativamente arbitraria y queda en gran medida a decisión del observador.
  3. Los enlaces internos de los componentes suelen ser más fuertes que los enlaces entre componentes. Este hecho tiene el efecto de separar la dinámica de alta frecuencia de los componentes (que involucra a la estructura interna de los mismos) de la dinámica de baja frecuencia (que involucra la interacción entre los componentes).



# Estrategias para atacar la Complejidad (2)

---

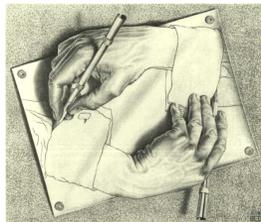
1. Los sistemas jerárquicos están compuestos usualmente de sólo unas pocas clases de subsistemas en varias combinaciones y disposiciones.
2. Un sistema complejo que funciona ha evolucionado de un sistema simple que funcionaba. Un sistema complejo diseñado desde cero nunca funciona y no puede “parcharse” para conseguir que lo haga. Hay que volver a empezar, partiendo de un sistema simple que funcione.



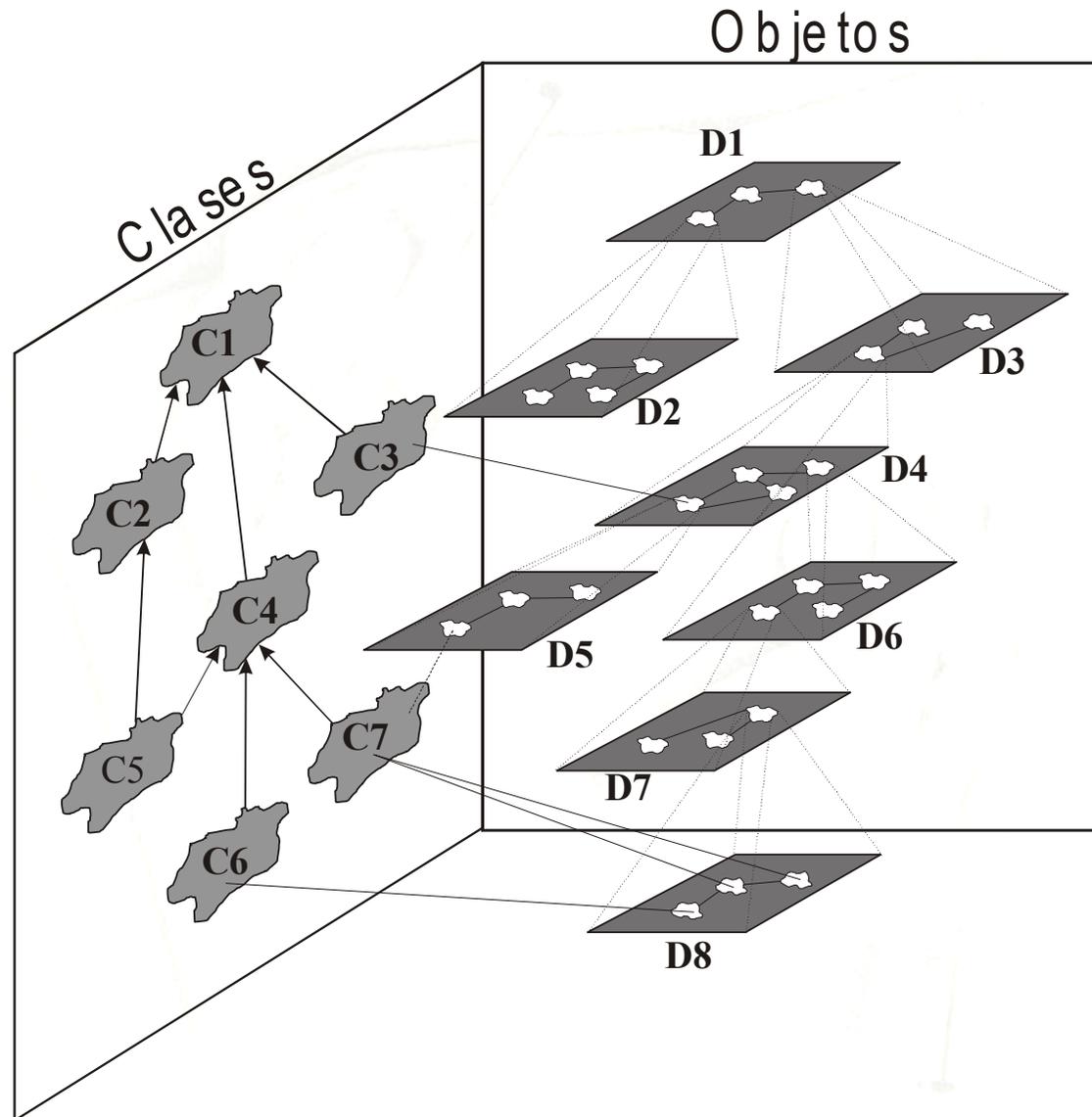
# Estrategias para atacar la Complejidad (3)

---

- Forma Canónica de un sistema complejo:
  - El descubrimiento de abstracciones y mecanismos comunes facilita en gran medida la comprensión de los sistemas complejos (piloto de avión).
  - La mayoría de los sistemas interesantes no contienen una jerarquía; en lugar de eso, suelen presentar varias jerarquías diferentes.
  - Por ejemplo, un avión puede estudiarse descomponiéndolo en su sistema de propulsión, sistema de control de vuelo, etc. Esta sería una jerarquía estructural o “**parte-de**”.
  - Pero puede diseccionarse el sistema por una vía completamente distinta. Por ejemplo un motor turboventilador es un tipo específico de motor de propulsión a chorro y un motor Pratt and Whitney TF30 es un tipo específico de motor turboventilador. Esta segunda jerarquía representa una jerarquía de tipos o “**es un**”.
  - Teniendo en cuenta las jerarquías “parte-de” y “es-un”, todos los sistemas complejos adoptan una misma forma: La **FORMA CANÓNICA**.



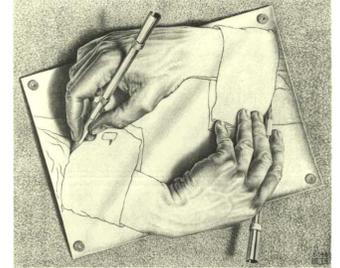
# Estrategias para atacar la Complejidad (4)



# Estrategias para atacar la Complejidad (5)

---

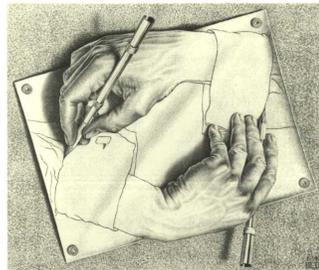
- Aquí se ven las dos jerarquías “ortogonales” del sistema:
  - La jerarquía “es-un” o su estructura de clases
  - La jerarquía “parte-de” o su estructura de objetos.
- Cada jerarquía está dividida en capas, con las clases y objetos más abstractos construidos a partir de otros más primitivos.
- La elección de las clases y objetos primitivos depende del problema que se maneja.
- Entre las partes de la estructura de objetos, existen colaboraciones estrechas entre objetos del mismo nivel de abstracción.
- Una mirada al interior de cualquier nivel dado revela un nuevo nivel de complejidad.



# Estrategias para atacar la Complejidad (6)

---

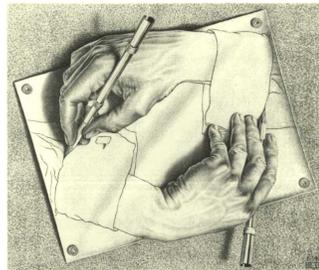
- La estructura de clases y la estructura de objetos no son del todo independientes: cada objeto de la estructura de objetos representa una instancia específica de alguna clase.
- Existen por lo general muchos más objetos que clases en un sistema complejo.
- La experiencia hace pensar que los sistemas complejos de software con más éxito son:
  - Aquellos cuyos diseños incluyen de forma explícita una estructura de clases y objetos bien construida.
  - Poseen los cinco atributos de los sistemas complejos que se describieron anteriormente.
- De forma conjunta, nos referimos a la estructura de clases y objetos de un sistema como su **arquitectura**.



# Estrategias para atacar la Complejidad (7)

---

- El problema para desarrollar sistemas de software con éxito:
  - Las limitaciones fundamentales de la capacidad humana para tratar la complejidad. Experimentos psicológicos, revelan que el máximo número de bloques de información que un individuo puede comprender de forma simultánea es de siete más menos dos.
- El papel de la descomposición:
  - “La técnica de dominar la complejidad se conoce desde tiempos remotos: *divide et impera* (divide y vencerás)” [Dijkstra, 1979].
  - Cuando se diseña un sistema de software complejo, es esencial descomponerlo en partes más y más pequeñas, cada una de las cuales se puede refinar entonces de forma independiente.

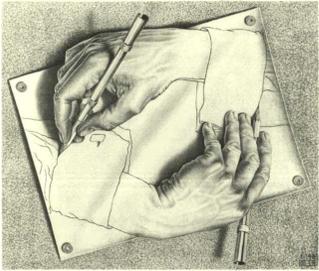


# Estrategias para atacar la Complejidad (8)

---

- Diferentes tipos de descomposición:

- **Descomposición algorítmica**



- Casi todos los informáticos hemos sido adiestrados en el dogma del diseño y/o programación estructurada descendente.
    - Cada módulo del sistema puede ser descompuesto en un conjunto de módulos más pequeños y manejables que realizan el proceso global.
    - No ahondaremos en el tema pues es parte de la formación de los primeros cursos del estudiante.

- **Descomposición orientada a objetos**

- Se puede descomponer un problema de acuerdo a las abstracciones claves de su dominio.
    - Estos agentes autónomos colaboran para llevar a cabo algún comportamiento de nivel superior, de manera que cada objeto contiene su comportamiento único y modela algún objeto del mundo real.

# Sistemas Software y Complejidad

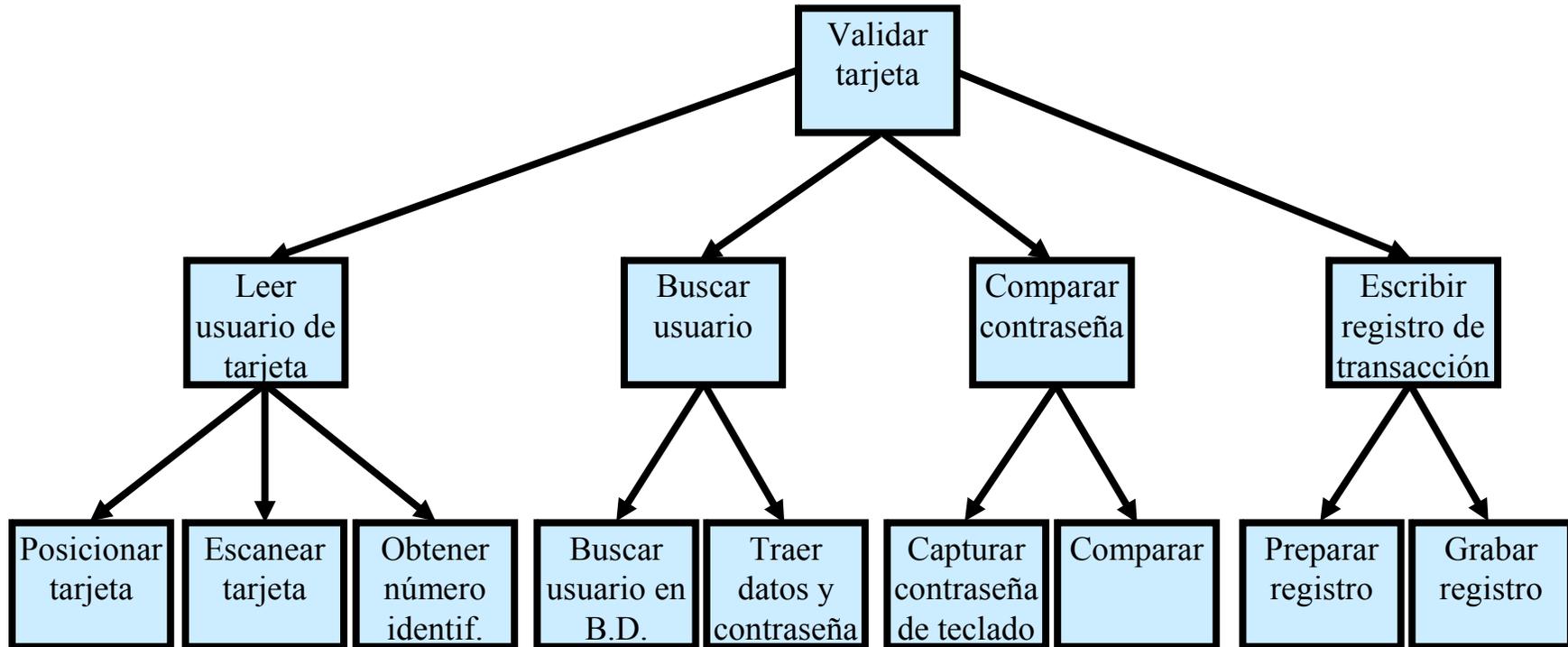
---

- Sistemas
  - Definición de sistema.
  - Sistemas Naturales y Sistemas Artificiales.
  - Sistemas de Computación.
  - Componentes Principales de cada Sistema.
- La complejidad
  - Principios Generales.
  - Estrategias para atacar la complejidad.



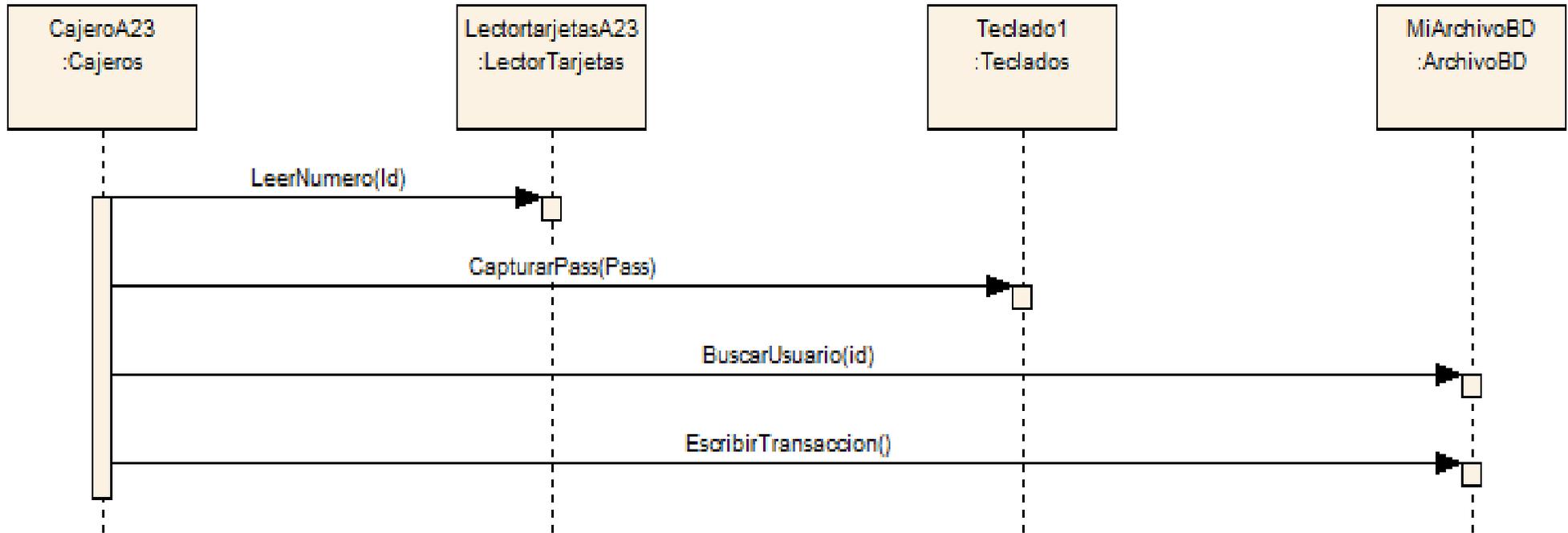
# Estrategias para atacar la Complejidad (9)

- **Descomposición algorítmica:** Validación de una tarjeta en un cajero automático.



# Estrategias para atacar la Complejidad (10)

- Descomposición orientada a objetos:



# Estrategias para atacar la Complejidad (11)

---

## Descomposición algorítmica versus descomposición orientada a objetos

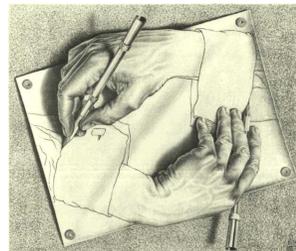
- Ambas visiones son importantes para el ingeniero de software, que debe aprender a usar las herramientas y debe saber cuándo y dónde usarlas.
- La visión algorítmica enfatiza el orden de los eventos y las funciones.
- La visión orientada a objetos resalta los agentes que causan y son sujetos de las acciones.
- No se puede construir un sistema complejo de las dos formas a la vez.
- La descomposición orientada a objetos (DOO) tiene una serie de ventajas altamente significativas sobre la descomposición algorítmica:
  - La DOO produce sistemas más pequeños a través de la reutilización de mecanismos comunes, proporcionando economía de expresión.
  - Los sistemas orientados a objetos son más resistentes al cambio y están mejor preparados para evolucionar en el tiempo (mantenimiento).
  - La DOO reduce en gran medida el riesgo que representa construir sistemas de software complejos y resuelve directamente la complejidad innata del software.



# Estrategias para atacar la Complejidad (12)

---

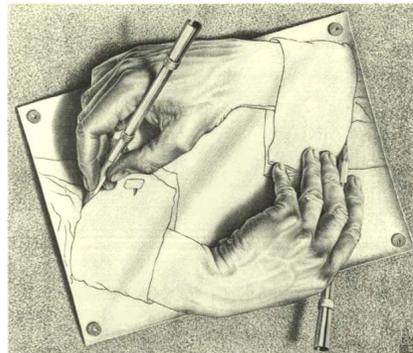
- El papel de la abstracción:
  - Los seres humanos hemos desarrollado una técnica excepcionalmente compleja para enfrentarnos a la complejidad: **realizamos abstracciones**.
  - **Abstracción**: ignoramos los detalles no esenciales de un sistema complejo y tratamos en su lugar con el modelo generalizado e idealizado del mismo.
  - Por ejemplo, cuando consideramos a una persona como un paciente de un consultorio, centramos nuestra atención en sus datos filiatorios, obra social, historia clínica y dejamos de lado el resto (su equipo de fútbol favorito, estudios cursados, partido político al que pertenece, etc.).
  - Los objetos, como abstracciones de entidades del mundo real, representan un agrupamiento denso y cohesivo de información que constituye bloques de alto contenido semántico, facilitando la comprensión de los sistemas.



# Estrategias para atacar la Complejidad (13)

---

- El papel de la jerarquía:
  - Las jerarquías de clases y objetos dentro de un sistema complejo me permiten incrementar el contenido semántico de los bloques de información.
  - La estructura de objeto permite comprender cómo diferentes objetos colaboran entre sí a través de interacciones llamadas mecanismos.
  - La jerarquía de clases resalta la estructura y comportamiento comunes en el interior de un sistema.
  - La identificación de jerarquías en un sistema complejo no suele ser fácil porque requiere que se descubran patrones entre múltiples objetos, cada uno con un comportamiento complicado.



# Sistemas Software y Complejidad

---

- Sistemas
  - Definición de sistema.
  - Sistemas Naturales y Sistemas Artificiales.
  - Sistemas de Computación.
  - Componentes Principales de cada Sistema.
- La complejidad
  - Principios Generales.
  - Estrategias para atacar la complejidad.

