

Datos, Sistemas de Información y Administración Estratégica

“Los **datos** consisten de hechos brutos, como número de empleado, número de horas laboradas en una semana, números de partes de inventario o pedidos de ventas. Cuando dichos hechos se organizan y ordenan en una forma significativa, se convierten en **información**. El **conocimiento** es la conciencia y comprensión de un conjunto de información y de las formas en que ésta puede ser útil para apoyar una tarea específica o llegar a una decisión”.

- “**Planeación estratégica**: Determinación de objetivos a largo plazo mediante el análisis de las fortalezas y debilidades de la organización, predicción de tendencias futuras y proyección del desarrollo de nuevas líneas de producto”.
- “La **toma de decisiones** es una parte vital de la administración estratégica de las empresas”.

Para que un negocio tenga éxito a nivel global, debe ser capaz de proporcionar la información correcta a las personas apropiadas en el momento oportuno, a pesar de que dichas personas se encuentren en cualquier parte del mundo. Las empresas pueden usar los sistemas de información con el fin de acrecentar sus ganancias y reducir sus costos.

Planeación estratégica para obtener una ventaja competitiva” se basa en analizar factores como liderazgo en costos, diferenciación, estructura de la industria y mejora de productos/servicios.

- “Un sistema es un conjunto de elementos o componentes que interaccionan con el fin de alcanzar un objetivo”
- “Un **sistema de información** (SI) es un conjunto de componentes interrelacionados que recaban, procesan, almacenan y distribuyen datos e información y proporcionan un mecanismo de retroalimentación para cumplir un objetivo”.

- “Los **componentes** de un sistema de información basado en computadora [CBIS] incluyen hardware, software, bases de datos, telecomunicaciones e internet, personal y procedimientos”.

“Los **tipos** de CBIS que las organizaciones utilizan pueden clasificarse en cuatro grupos básicos”:

- 1) comercio electrónico y comercio móvil.
- 2) sistemas TPS y ERP.
- 3) sistemas MIS y DSS.
- 4) sistemas de información especializados en negocios.

Entradas, procesamiento, salidas y retroalimentación

Entrada: Actividad de recabar y capturar datos.

Procesamiento: Conversión o transformación de los datos en salidas útiles.

Salida: Producción de información útil, por lo general en forma de documentos y reportes.

Retroalimentación: Salida que se utiliza para realizar cambios en la entrada o en las actividades de procesamiento.

El **objetivo del sistema de información** debe capturar tanto "lo que el sistema pretende lograr" como la manera en que contribuirá al proceso de toma de decisiones dentro de la organización.

- “El sistema de información tiene como objetivo principal registrar, procesar y brindar acceso a la información necesaria para [nombre del proceso o necesidad del negocio agrupado por gestiones], permitiendo [beneficio esperado], y facilitando la toma de decisiones [operativas / de control / directivas] en [nombre de la organización].”

- “El **límite** de un sistema de información está compuesto por la **primera** y **última** función que el sistema cubre dentro del ciclo de procesos que se busca soportar.”
- **Alcance:** Para estructurar el alcance de manera clara y precisa, es recomendable utilizar verbos en infinitivo, que describan las acciones concretas que el sistema realizará.

Nivel	Objetivo general	Gestión integrada	
	Objetivo específico (req. Global)	Verbo	(Req. Detallado)
Esencial	Gestionar / Administrar	Registrar	Alta de un objeto o Transacción esencial
		Modificar	Modificación de un objeto o Transacción esencial
		Eliminar	Eliminar un objeto o Transacción esencial
		Consultar	Consultar información de un objeto esencial o de soporte.
		Emitir	Emitir un listado de información de uno o más objetos. Ejemplo: Emitir listado de clientes.
		Generar	Generar un objeto como resultado de un proceso que combina uno o más objetos. Ejemplo: Generar estadísticas de venta de productos.
Soporte		Actualizar	(Registrar, modificar, eliminar y consultar)

Ingeniería de Software

"La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales."

1) Capas de la ingeniería del software

La ingeniería del software se estructura en varias **capas**, cada una con un rol clave para garantizar la calidad del producto. En la base de todo está la **capa de proceso**, que actúa como una guía general que organiza y coordina el trabajo. Sobre esta capa se apoyan tres dimensiones clave:

1. **Herramientas:** son los programas o sistemas que automatizan parte del trabajo.
2. **Métodos:** son los enfoques técnicos para analizar, diseñar, construir y probar el software.
3. **Proceso:** El proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software.
4. **Enfoque de calidad:** abarca prácticas que aseguran que el producto cumpla con estándares y sea confiable, útil y mantenible.

Estas capas trabajan juntas para que el desarrollo de software sea sistemático y profesional.

2) Proceso de software

Comunicación: “Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente. Se busca entender los objetivos de los participantes respecto del proyecto”

Planeación: “La actividad de planeación crea un ‘mapa’ que guía al equipo mientras viaja define el trabajo de ingeniería de software al describir tareas técnicas, riesgos, recursos, productos y programación.”

Modelado: “Un ingeniero de software crea modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará.”

Construcción: “Esta actividad combina la generación de código y las pruebas que se requieren para descubrir errores.”

Despliegue: “El software se entrega al consumidor que lo evalúa y que le da retroalimentación”

Modelado extendido (Patrones en la construcción del modelo conceptual)

“Los **patrones de software** son un mecanismo para capturar conocimiento del dominio, en forma que permita que vuelva a aplicarse cuando se encuentre un problema nuevo.”

“Una vez descubierto el patrón, se documenta describiendo explícitamente el problema general al que es aplicable el patrón, la solución prescrita, las suposiciones y restricciones, ventajas y desventajas del mismo y ejemplos de su empleo en aplicaciones prácticas.”

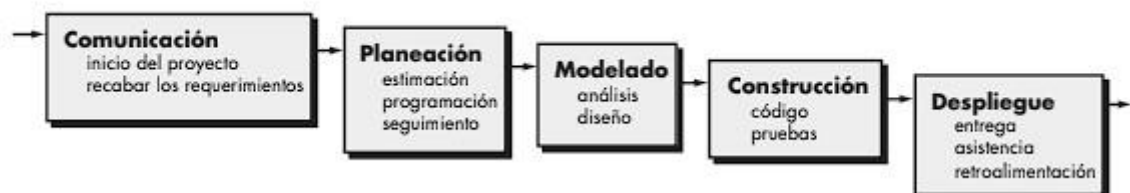
Tipos de Modelos de Proceso de Software

Los **modelos de proceso** representan diferentes formas de organizar el trabajo de desarrollo de software. Cada uno responde a necesidades y contextos distintos:

1. Modelo en Cascada (Waterfall)

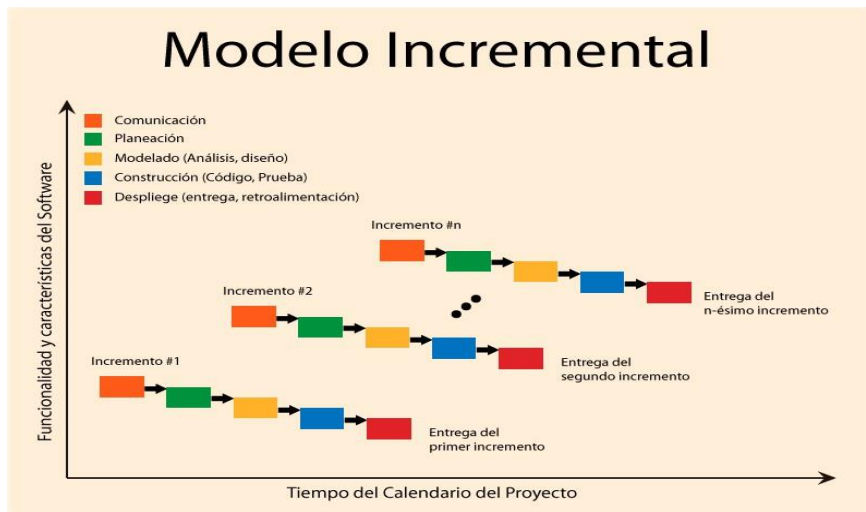
- Se basa en fases secuenciales: análisis, diseño, codificación, pruebas y mantenimiento.
- Cada etapa debe completarse antes de pasar a la siguiente.
- Es rígido y adecuado sólo si los requerimientos están bien definidos desde el principio.

Modelo de la cascada



2. Modelo Incremental

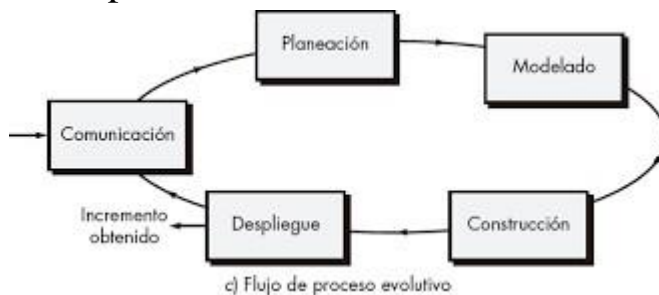
- Se construye el software en **partes funcionales** (incrementos).
- Cada entrega suma nuevas capacidades hasta completar el sistema.
- Mejora la retroalimentación temprana y permite ajustar el desarrollo con mayor flexibilidad.



3. Modelos Evolutivos

Incluyen variantes como:

- **Prototipado:** se desarrolla un modelo preliminar que ayuda a entender mejor los requerimientos.
- **Modelo Espiral:** combina iteración con evaluación de riesgos. Es útil para proyectos grandes y complejos.
- **Desarrollo Concurrente:** permite trabajar en varias actividades al mismo tiempo y facilita cambios durante el proceso.



4. Modelos Ágiles

- Priorizan la adaptabilidad, la colaboración con el cliente y la entrega rápida y continua de valor.
- Se trabaja en **iteraciones breves** (por ejemplo, sprints en Scrum).
- Ejemplos: **Scrum, Extreme Programming (XP), Lean Software Development.**

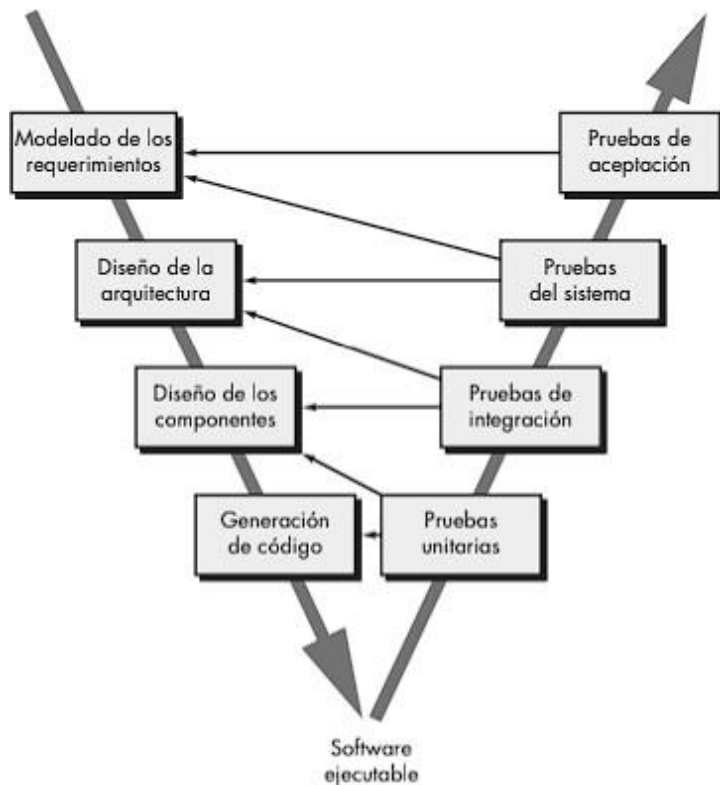


5. Modelos Especializados

- **Desarrollo basado en componentes:** reutiliza módulos de software preexistentes.
- **Métodos formales:** usan matemáticas para especificar y verificar programas críticos (ej: en medicina o aviación).
- **Orientado a aspectos:** se enfoca en separar funcionalidades transversales como seguridad o logging.

6. Modelo en V

- Es una variante del modelo en cascada, pero pone énfasis en la verificación y validación tempranas.
- Cada fase de desarrollo tiene asociada una fase de pruebas correspondiente.
- Representa el ciclo como una “V”: o Lado izquierdo: análisis y diseño. o Lado derecho: pruebas que verifican cada fase previa.
- Muy usado en sistemas críticos, donde los errores pueden tener consecuencias graves (como control aéreo o medicina).
- Promueve una alta rigurosidad documental y de calidad.



Ingeniería de Requerimientos

La ingeniería de requerimientos es una **actividad fundamental de la ingeniería de software**, porque todo sistema bien construido comienza con una comprensión precisa de lo que debe hacer, para quién y bajo qué condiciones. En contextos empresariales, esto es vital para que herramientas como ERP y CRM respondan realmente a las necesidades organizativas.

- Un **requerimiento** es una condición o capacidad que debe ser satisfecha por un sistema o componente para resolver un problema del negocio o alcanzar un objetivo.
- En el mundo empresarial, los requerimientos no son solo “funcionalidades técnicas”, sino representaciones directas de procesos de negocio (como facturación, logística, atención al cliente).
- En herramientas como *SAP o Dynamics 365*, los requerimientos guían cómo se configura el sistema: qué módulos se activan, qué flujos se automatizan, qué restricciones se imponen.

2. Clasificación de los Requerimientos

Requerimientos Funcionales (RF): Son descripciones **explícitas** de lo que el sistema debe hacer: tareas, servicios, respuestas a entradas.

Características:

- Son observables: permiten validar si el sistema los cumple.
- Se expresan típicamente en forma de “El sistema deberá...”

En un *CRM*, un requerimiento funcional podría ser: “Permitir asignar clientes potenciales a representantes según zona geográfica”.

¿Cómo identificar un Requerimiento Funcional (RF)?

Me fijo si describe una acción específica que el sistema debe realizar, como:

- Registrar, consultar, modificar, eliminar, generar, emitir, etc.
- Una entrada → una respuesta concreta.
- Es algo que **el usuario puede hacer directamente** o que el sistema **debe ejecutar como parte de su flujo principal**

Requerimientos No Funcionales (RNF): Definen **cómo debe comportarse** el sistema, incluyendo limitaciones, calidad, rendimiento, seguridad, usabilidad.

¿Cómo identificar un Requerimiento No Funcional (RNF)?

👉 **Me doy cuenta cuando habla de cómo** se comporta el sistema, no **qué** hace.

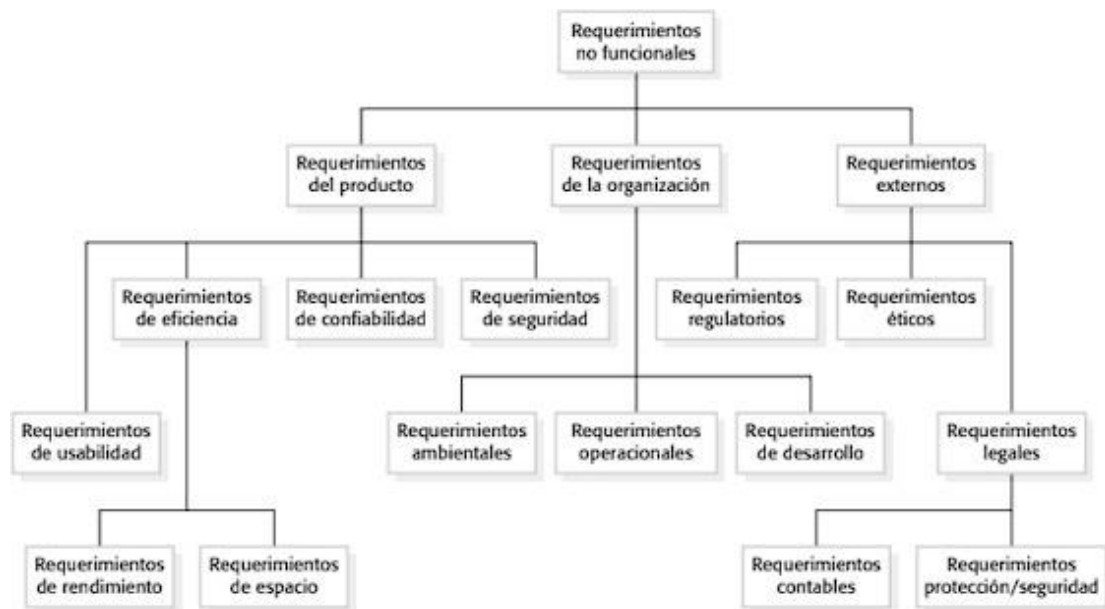
Son condiciones, restricciones o cualidades del sistema.

🔍 Pistas claras:

- No hay una acción directa, sino un enfoque en:
 - Tiempos (“en menos de X segundos”) → rendimiento.
 - Seguridad, confiabilidad, disponibilidad, usabilidad.
 - Normas técnicas, compatibilidad, accesibilidad.

A menudo vienen como **recomendaciones, condiciones técnicas o atributos de calidad**.

Tipos de RNF:



3. Especificación de Requerimientos

La **especificación** es el proceso de traducir esos requerimientos en documentos formales y detallados, que sirven como referencia para desarrollo, pruebas y validación.

- **Requerimientos del usuario:** redactados en lenguaje natural, orientados a usuarios clave (gerentes, stakeholders, clientes).
- **Requerimientos del sistema:** técnicos, específicos, detallados, para programadores y diseñadores.

4. El Proceso de Ingeniería de Requerimientos

Somerville lo estructura como un proceso iterativo compuesto por las siguientes etapas:

4.1 Elicitación (Adquisición)

Consiste en recolectar la información clave:

- Entrevistas.
- Cuestionarios.
- Observación de procesos.
- Prototipos.

4.2 Análisis

Una vez obtenida la información, se analizan:

- Conflictos entre requerimientos
- Priorización (no todo puede hacerse al mismo tiempo)
- Detección de ambigüedades

4.3 Especificación

Elaboración del *documento de requerimientos* o SRS (Software Requirements Specification), que puede incluir:

- Diagramas UML
- Casos de uso
- Tablas de reglas de negocio

- Matrices de trazabilidad

4.4 Validación

Confirmar con los usuarios que:

- Los requerimientos reflejan lo que necesitan.
- No hay malentendidos o suposiciones erróneas.

5. Administración de Requerimientos

Los requerimientos cambian (¡siempre!). La gestión de cambios es crucial para evitar caos:

- Versionado
- Registro de solicitudes
- Trazabilidad hacia pruebas, código, documentación

6. Importancia en la Organización Empresarial

Toda organización es un conjunto de procesos interrelacionados. Cuando implementamos un sistema ERP o CRM, *automatizamos y digitalizamos esos procesos*.

Una mala definición de requerimientos puede causar:

- Flujos innecesarios o redundantes
- Costos extra por personalización incorrecta
- Pérdida de confianza del usuario final
- Fracaso del proyecto