

# Qué es la Ingeniería de Software

- Surgimiento
- Definiciones
- Solución de Problemas
- Relación con otras disciplinas
- El Software en la sociedad actual
- Ética y Responsabilidad
- Calidad del Software
- Participantes en el Desarrollo
- Ingeniería de Software e Ingeniería de Sistemas
- Cambios y Tendencias

# *Surgimiento*

- En los comienzos:
  - el programador era el usuario
  - los problemas a resolver bien conocidos y simples
- Desarrollo de las computadoras:
  - aparece la figura del programador especializado
  - la mayor potencia de los equipos permite atacar problemas más complejos
  - Se habla de la "Crisis del Software"
- El término Ing. de Software aparece por 1ra. vez en 1968

# *Definiciones*

- Software: Programas de computador, procedimientos, y la documentación y los datos posiblemente asociados relacionados con la operación de un sistema de computador - IEEE 90.
- ...(el ingeniero) aplica el método y enfoque científico a la solución de problemas...
- Ingeniería de Software:(1)aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, esto es, la aplicación de la ingeniería al software -  
(2)El estudio de enfoques como en (1) - IEEE 90.

# *Solución de Problemas*

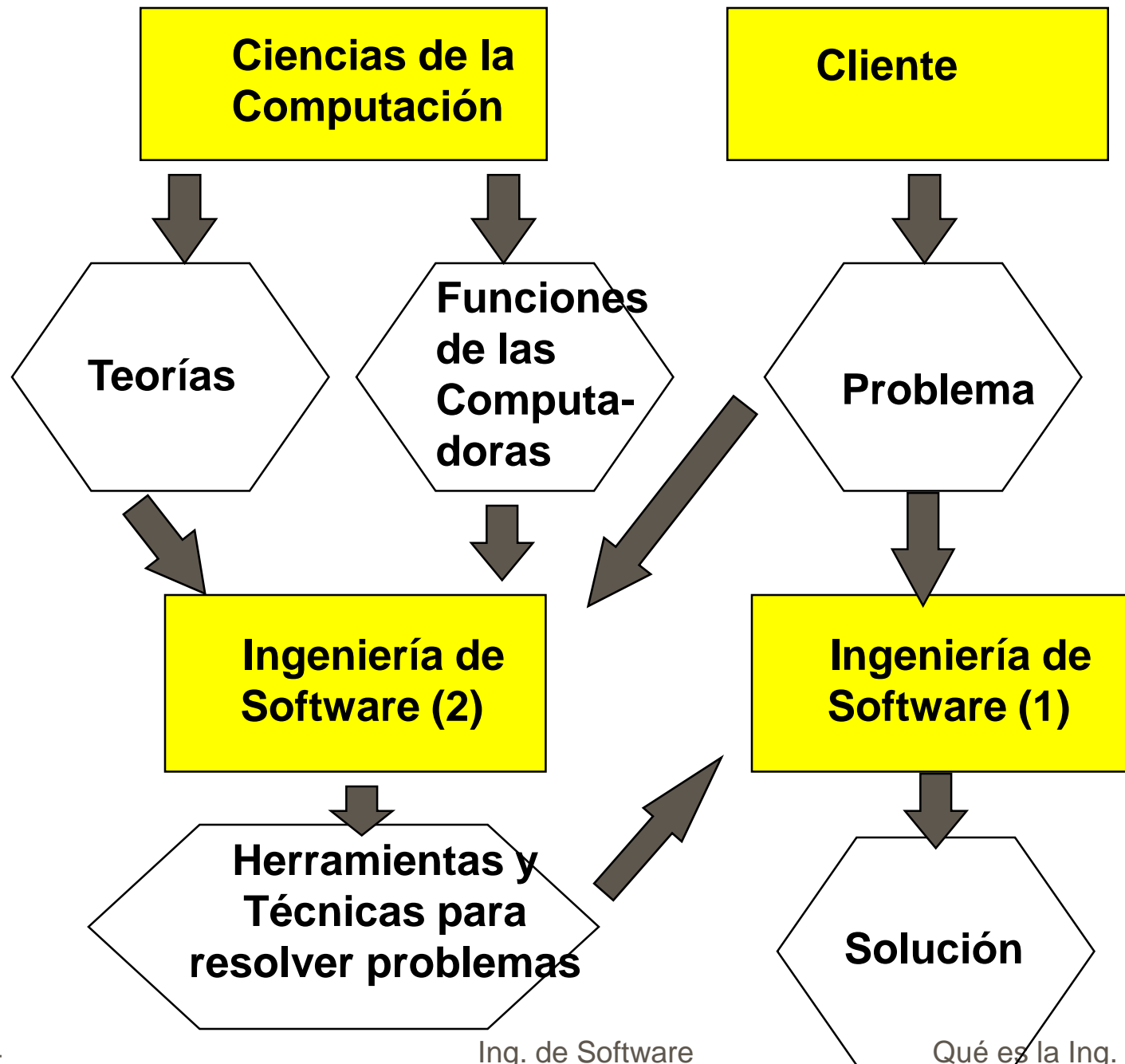
- Solucionar un problema complejo
  - **Análisis** (en subproblemas más simples)
  - **Síntesis** de una solución al problema a partir de las soluciones a los subproblemas
- Recurriendo a:
  - Técnicas, Métodos
  - Herramientas (mejorar resultado o eficiencia)
  - Procedimientos (combinan métodos y herramientas)
  - Paradigmas (“estilo” de la solución)

# Definiciones

- **Técnica**
  - Procedimiento técnico para construir o evaluar software que requiere habilidades y produce un resultado técnico, p.e. probar, codificar, etc.
- **Método**
  - Es un procedimiento para gestionar la aplicación de técnicas de software (cuándo y cómo aplicar, cuándo comenzar y terminar, qué técnica es apropiada, cómo evaluarla), p.e. plan de prueba, recopilar requerimientos
- **Herramienta**
  - Producto de software que facilita la ejecución de una técnica o método
- **Procedimiento**
  - Combinación de métodos, técnicas y herramientas
- **Paradigma**
  - Filtro conceptual que determina cómo percibimos/interpretamos

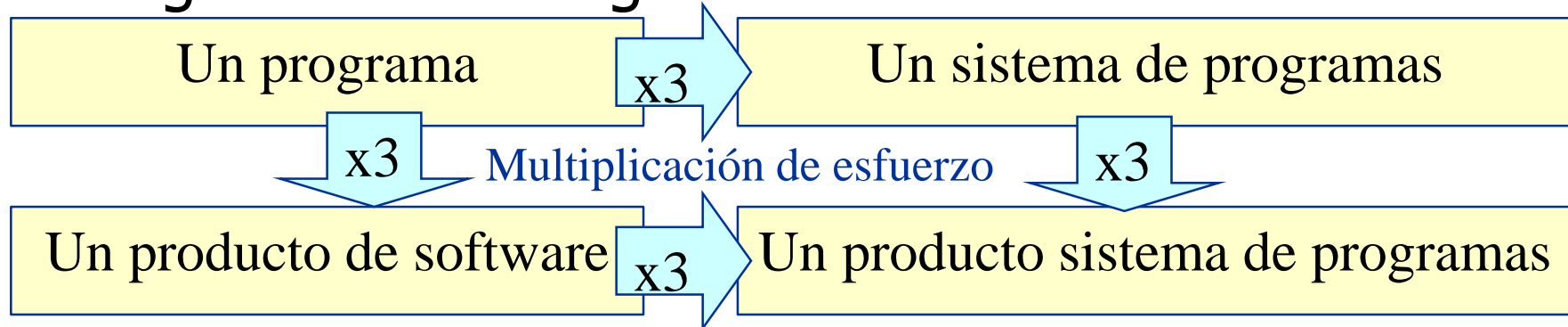
# *Relación con otras disciplinas*

- Ciencias de la Computación aporta:
  - Teorías
  - Funciones de Computadoras
- Cliente trae:
  - Problema
- Ingeniería de Software (2) desarrolla:
  - Métodos, Herramientas, Procedimientos, Paradigmas para resolver problemas
- Ingeniería de Software (1) resuelve problemas



# Relación con otras disciplinas

- Programación -Programa-Producto-Sistema



- Ingeniería: “Construcción de artefactos cumpliendo con restricciones”

- Arte más que Ciencia (hacer más que saber)
- Plazo, Costo, Calidad, Otras

- Administración

- Proyectos, Recursos Humanos



# *Software en la Sociedad Actual*

- ¿Dónde no está presente?
  - Energía
  - Comunicaciones
  - Automóviles
  - Electrodomésticos
  - Equipos médicos
  - ...
- ¿ Crisis del Software?

# *Etica y Responsabilidad*

- Repercusiones de fallas en el software:
  - pérdidas financieras
  - riesgo a la seguridad
  - ...
- Más allá de las fallas:
  - Impacto social
  - Calidad de vida
  - Cuestiones legales

# *Ing. de SW ¿una profesión?*

- SWEBOK – IEEE 2004
  - Software Engineering Body of Knowledge
  -
- Código de Ética – ACM – IEEE-CS
  - En su versión corta establece...

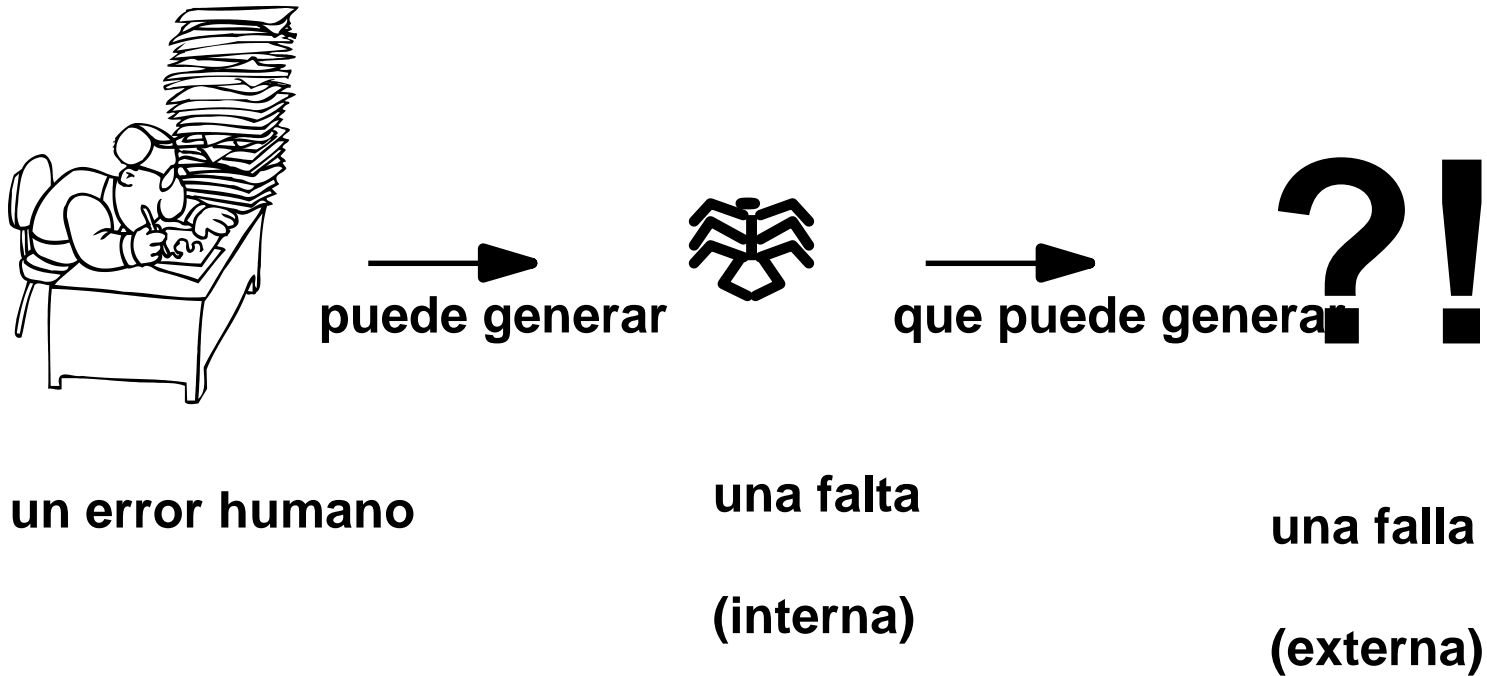
# *Los ingenieros de software...*

- actuarán en forma congruente con el **interés social**
- actuarán de manera que se concilien los mejores intereses de sus **clientes y empresarios**, congruentemente con el **interés social**
- asegurarán que sus productos y modificaciones correspondientes cumplen los **estándares profesionales más altos posibles**
- mantendrán **integridad e independencia** en su juicio profesional
- gerentes y líderes promoverán y se suscribirán a un **enfoque ético** en la administración del desarrollo y mantenimiento de software
- incrementarán la **integridad y reputación de la profesión** congruentemente con el interés social
- apoyarán y serán justos con sus **colegas**
- participarán **toda su vida** en el **aprendizaje** relacionado con la práctica de su profesión y promoverán un **enfoque ético** en la práctica de la profesión

# *Calidad*

- Problemas:
  - Plazo
  - Costo respecto a presupuesto
  - Utilidad
  - Requerimientos oscuros o cambiantes
  - Fallas
  - Rigidez
  - Alto costo de mantenimiento
  - Riesgos

# Calidad (falta-falla)



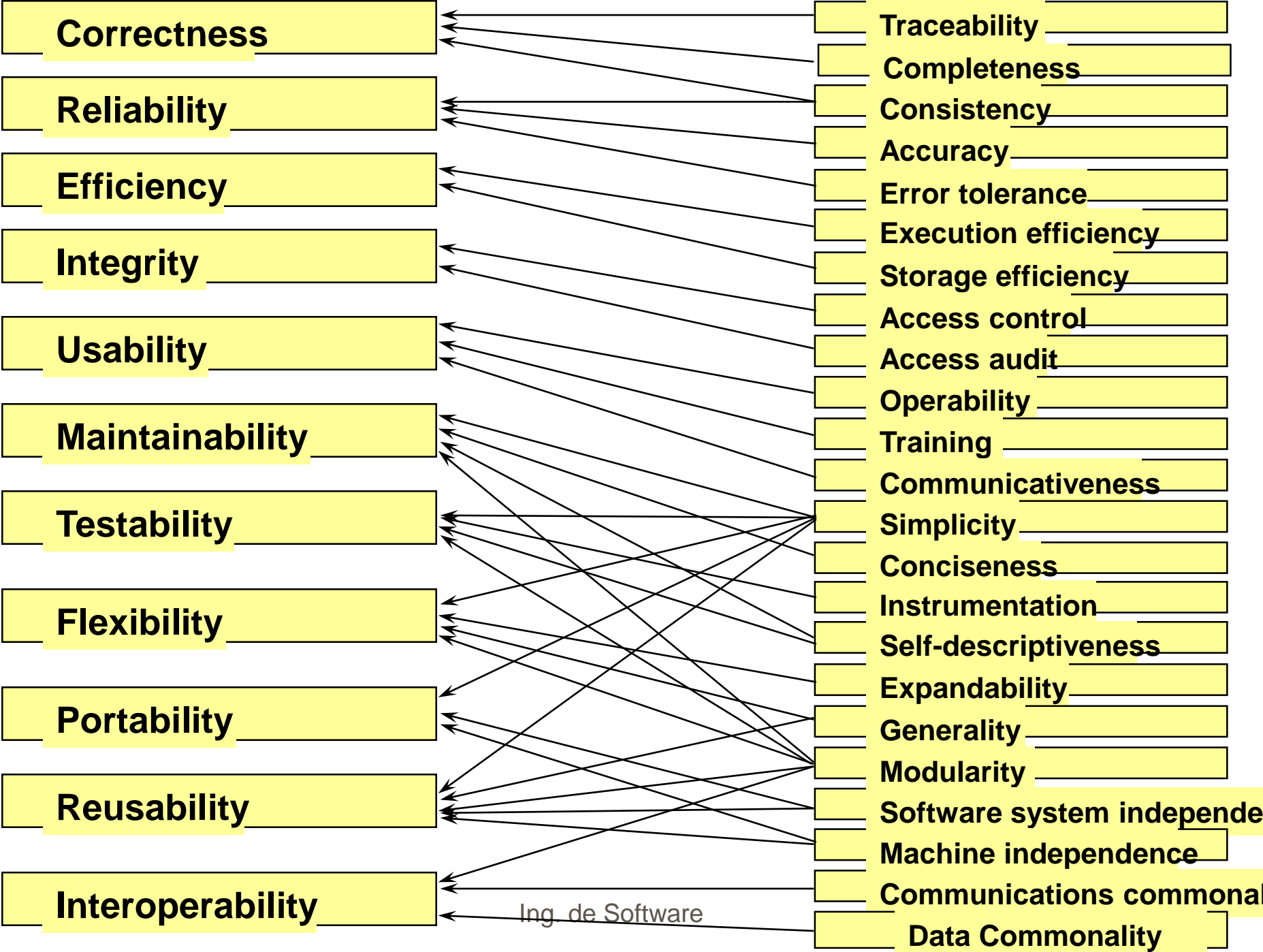
# Calidad - Visiones

- ¿Qué es la Calidad?
- Visión:
  - Trascendente -se reconoce pero no se puede definir
  - del Usuario - adecuación al uso
  - del Productor- adecuación a las especificaciones
  - del Producto - características específicas
    - **comportamiento externo (visible para todos)**
    - **características internas (normalmente sólo visibles al productor)**
  - basada en el Valor - cuánto estaría dispuesto a pagar

## *Calidad - Visiones(2)*

- **Usuario:**
  - satisfacer necesidades/expectativas (utilidad, tiempo de respuesta)
  - esfuerzo necesario (facilidades de aprendizaje y uso)
  - sin inconvenientes (frecuencia e impacto de fallas)
- **Implementador:**
  - cantidad y tipo de faltas
  - facilidad de entender
  - bajo impacto de las modificaciones





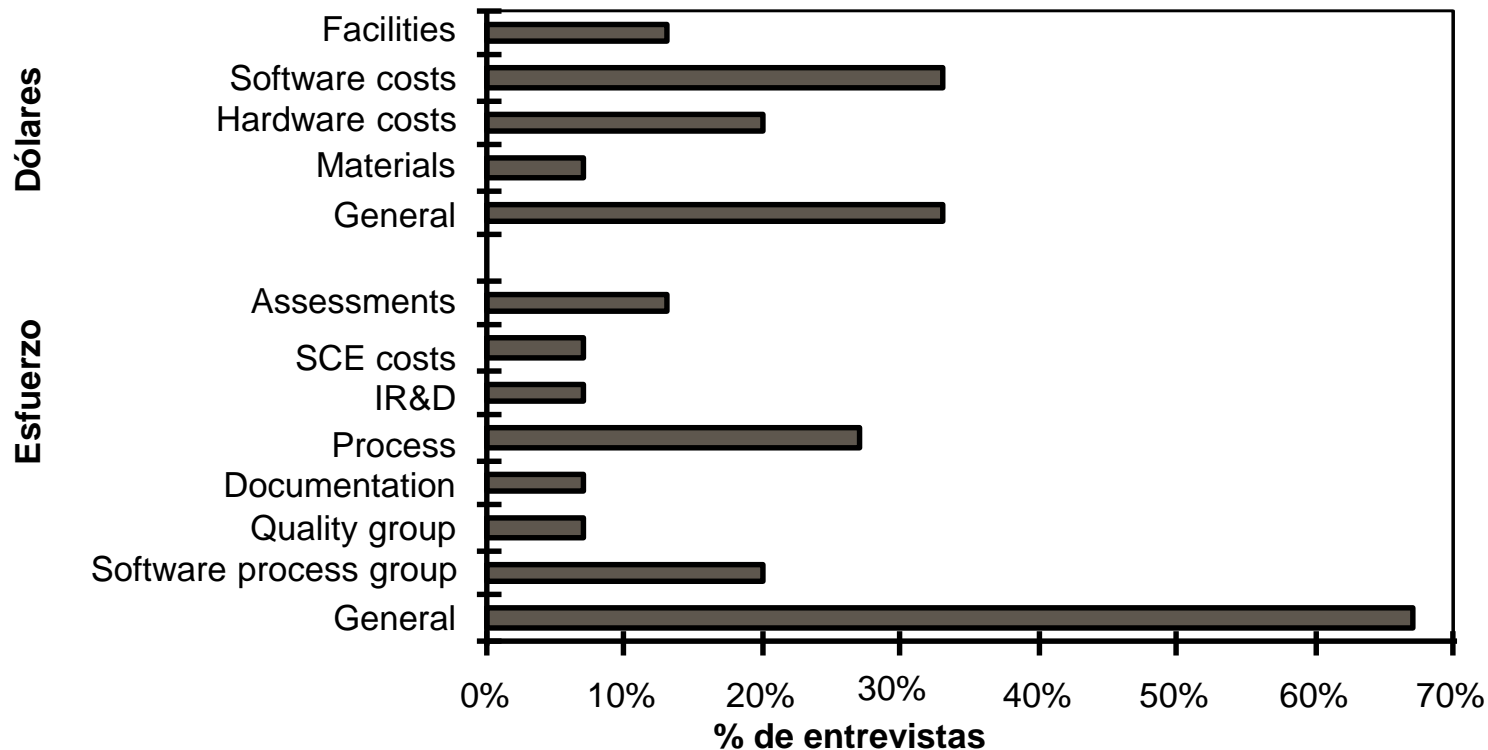
## *Calidad - Visiones (4)*

- según la visibilidad:
  - factores externos (visibles a todos)
  - factores internos (visibles a los implementadores)
- atinentes al:
  - Producto (una vez que el producto ya existe)
  - Proceso de Producción (mientras se produce)
- Qué relación habrá entre
  - ¿factores internos y externos?
  - ¿factores del proceso y del producto?

## *Calidad - Visiones(5)*

- En el contexto del Negocio
  - Valor Técnico - Valor para el Negocio
- Retorno de la Inversión (Negocio)
  - Período de Repago
  - Tasa de Retorno
  - Valor Actual de los Flujos de Caja
- Retorno de la Inversión (Software)
  - Factores de Costo, Beneficios

# Calidad - Valor para el negocio



**Disparidad de factores considerados en la definición de ROI en SW entre gobierno de US y sus proveedores**

# *Quiénes participan*

- Desarrollo a Medida
  - Cliente (solicita)
  - Desarrollador construye
  - Usuario
- COTS (Commercial Off The Shelf)
- Desarrollador subcontrata parte
- Producto “llave en mano”
- Personalización
- Integración

**CLIENTE**



**Patrocina el desarrollo del sistema**

\$\$\$,  
necesidades

**USUARIO**



**Usa el sistema**

**DESARROLLADOR**

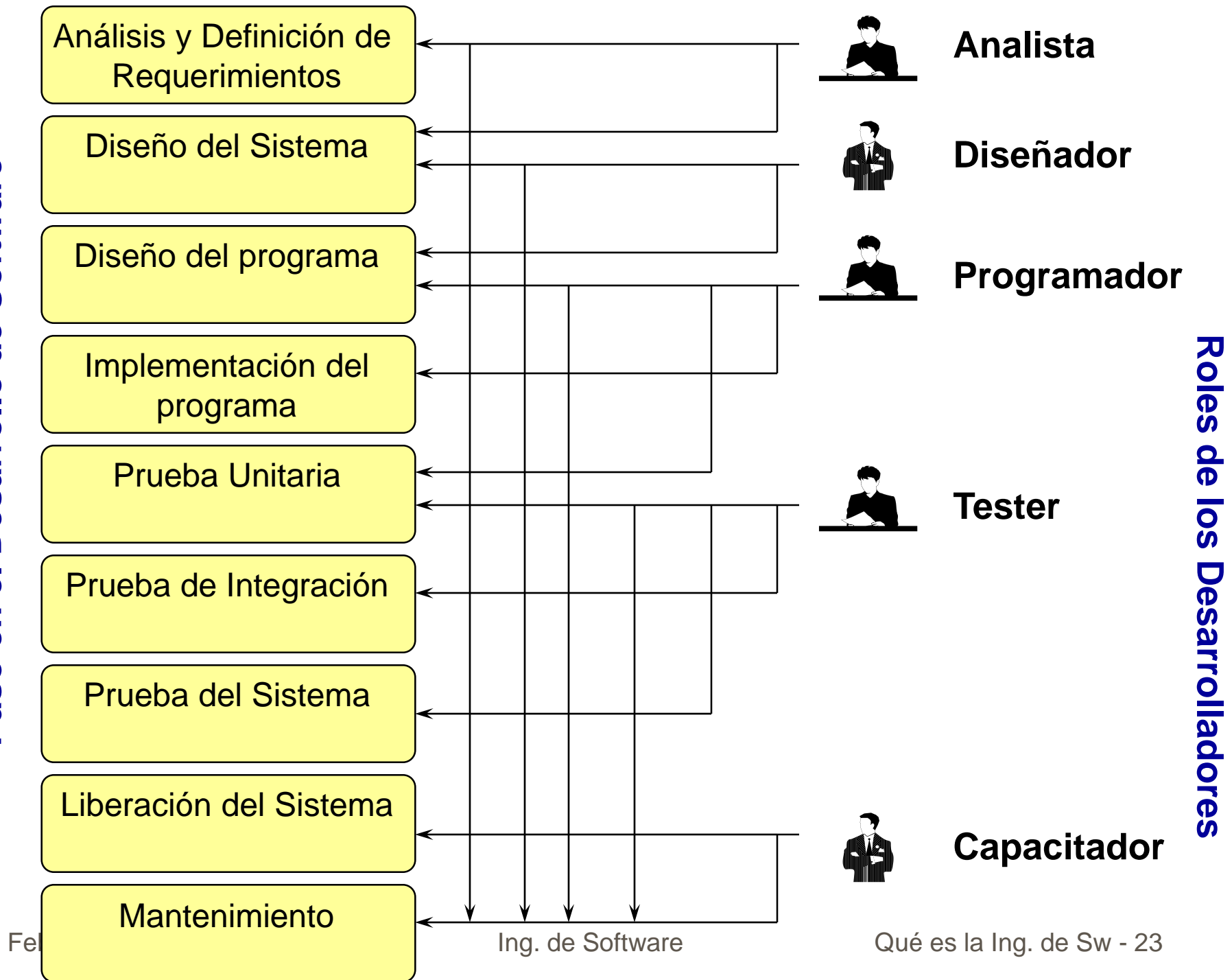


**Construye el sistema**

Obligación contractual

Necesidades

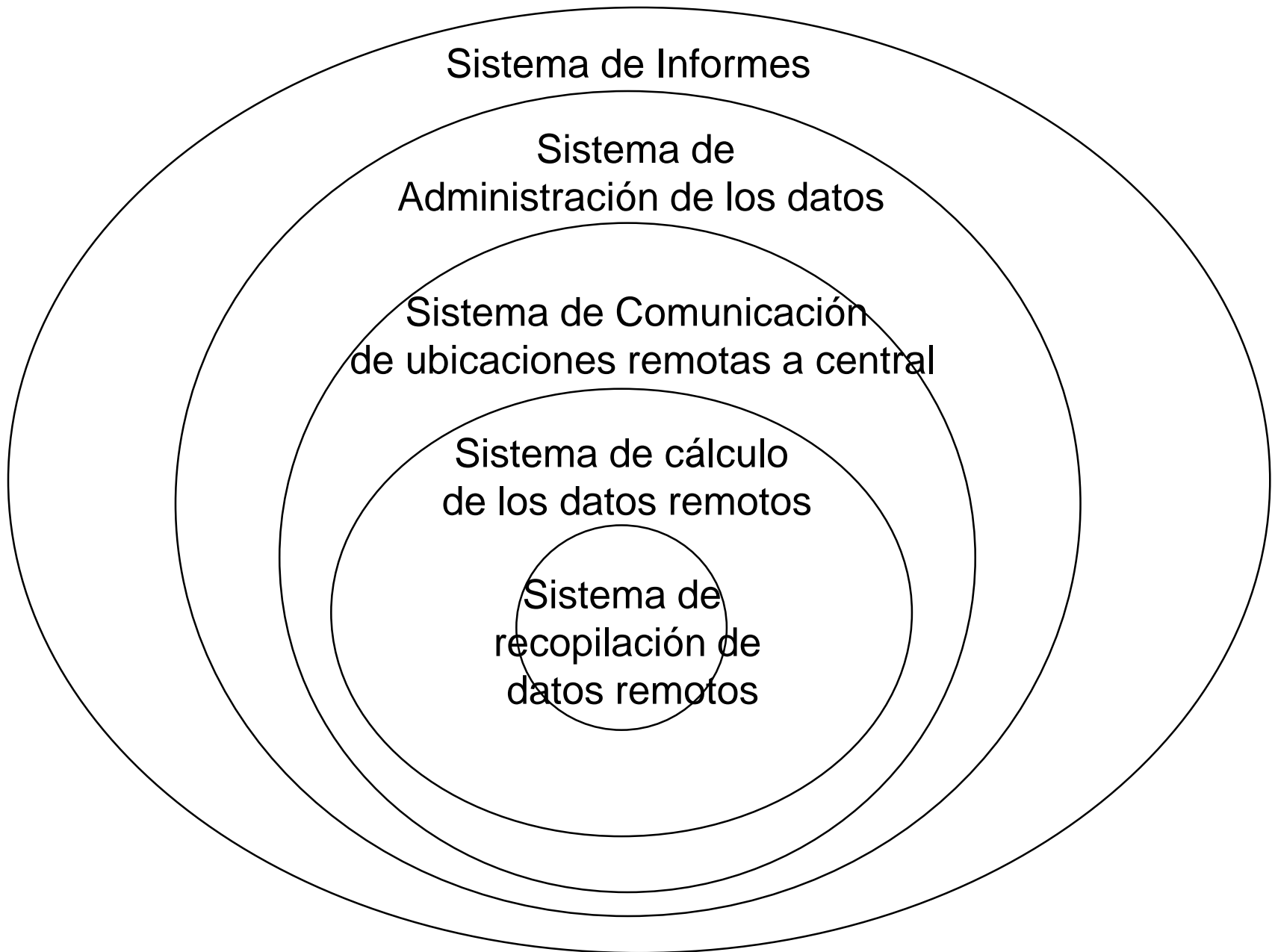
Sistema de software



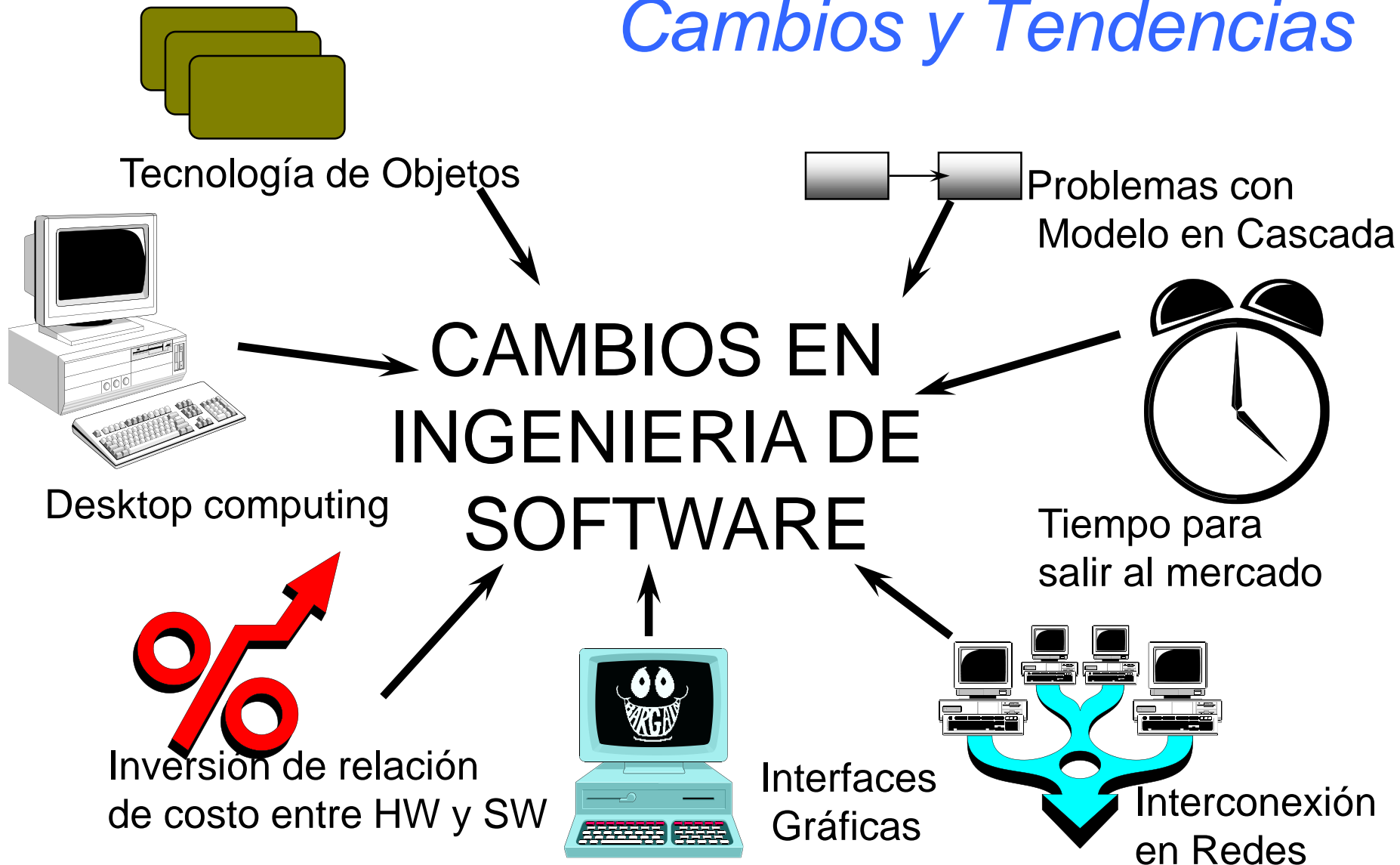
# *Ing. Software y de Sistemas*

- El Software como componente de un sistema:
  - Hardware
  - Software
  - Personas
- Sistema
  - Límite
  - Interacción con el exterior
  - Componentes y sus relaciones





# *Cambios y Tendencias*



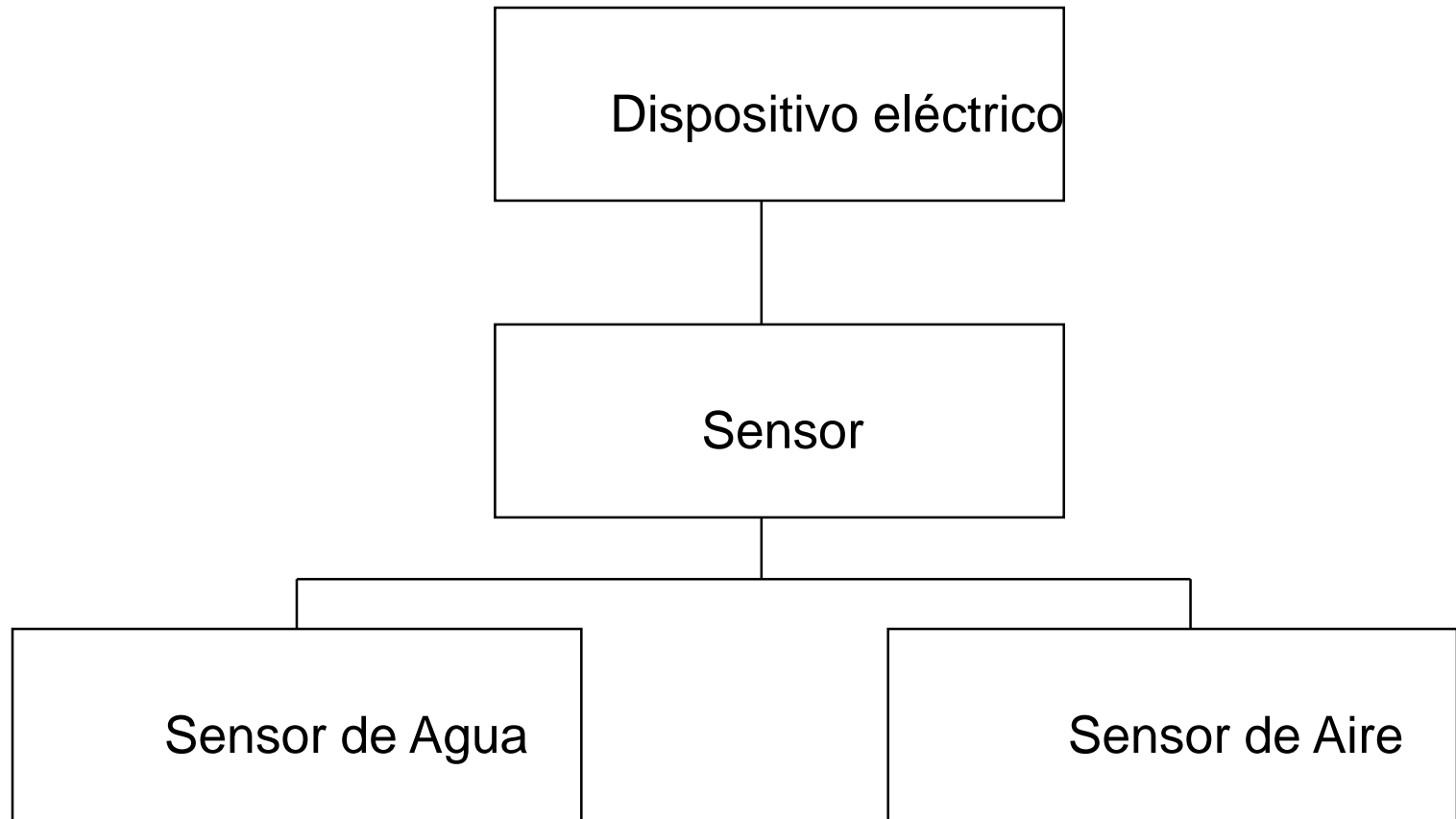
# *Cambios y Tendencias*

- Nociones Fundamentales:
  - Abstracción
  - Métodos y Notaciones para Análisis y Diseño
  - Prototipación
  - Arquitectura del Software
  - Proceso de Software
  - Reuso
  - Mediciones
  - Herramientas y Ambientes Integrados

# *Abstracción*

- es una descripción del problema a cierto nivel de generalización que permite concentrarnos en los aspectos esenciales del problema sin preocuparnos de detalles.
- Identificar clases de objetos que permiten agrupar elementos
- Formar jerarquías

# *Abstracción*



# *Métodos de Notación Análisis y Diseño*

- Construir Modelos y verificar completitud y consistencia
- Usar notación estándar para ayudarnos a comunicar y documentar las decisiones

# *Prototipos*

- Construir una pequeña versión de un sistema para
  - ayudar al usuario y al cliente a identificar los requerimientos clave
  - demostrar la factibilidad de un diseño o enfoque
- ¿El usuario se siente cómodo con la apariencia y forma de interacción?

# *Arquitectura*

- La arquitectura de un sistema lo describe en términos de un conjunto de unidades arquitectónicas y de cómo esas unidades se relacionan entre sí
- Existen distintos “estilos arquitectónicos” y distintos enfoques para identificar las unidades arquitectónicas



# *Arquitectura- cómo descomponer*


- a partir de las funciones: asignar funciones a módulos
- orientada a los datos: a partir de las estructuras de datos externas
- orientada a eventos
- diseño de afuera hacia adentro: basada en las entradas del usuario al sistema
- diseño orientado a objetos: basado en identificar clases de objetos y sus interrelaciones

# *Proceso*


- Proceso de desarrollar software (organización y disciplina en las actividades)
- contribuir a la calidad del software y a la velocidad con la que se desarrolla

# Proceso- especificidad

**Desarrollo  
Controlado**



**Desarrollo  
Rápido  
de aplicaciones**



**Aplicaciones para  
toda la  
empresa o división**

- Mision -critica
- Multi -usuario
- Multi -plataforma
- 2- 3-capas

**Aplicaciones  
Departamentales**

- Visión/Alcance limitados
- Riesgo bajo/medio
- mono/ multi -plataforma
- 1-2-capas

**Mono-usuario,  
herramientas para  
PC de escritorio**

- Paquetes/desarrollo minimo
- Bajo costo/bajo riesgo
- plataforma única

# *Reuso*

- Sacar partido de los elementos comunes entre aplicaciones reutilizando elementos de desarrollos previos
- Componentes Reusables como bienes de activo del negocio

## *Mediciones*

- Cuantificando dónde estamos y lo que podemos, describimos nuestras acciones y sus resultados en un lenguaje matemático común que nos permite evaluar nuestro progreso.

# *Herramientas y Ambientes Integrados*

- Utilizar herramientas para mejorar el desarrollo de software, ayudan en:
  - análisis y diseño
  - registrar el avance
  - depurar programas
  - probar programas

# Preguntas

- Brinde al menos una definición de Ing. de software (IS)
- ¿Cuántos años tiene como disciplina?
- ¿Con qué otras disciplinas está relacionada?
- ¿Qué se entiende por “crisis del software”?
- ¿Cómo afecta a la IS que el software esté presente en las más diversas áreas?
- ¿Qué implicancias tiene sobre la ética y responsabilidad?
- ¿Qué relación tiene la IS con cuestiones legales?

## *Preguntas (2)*

- ¿Cuál es la diferencia entre falta (o defecto) y una falla del software?
- ¿Cuándo es posible evaluar los factores de calidad externos de un producto de software?
- ¿Qué problemas plantea la evaluación de los factores de calidad externos durante el proceso de desarrollo?
- ¿Qué nociones fundamentales han persistido?