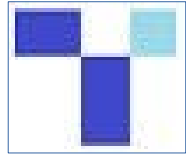


Introducción a la Ingeniería de Software



Tema 1:

“Qué es la Ingeniería de Software?”

Agenda

- Surgimiento
- Definiciones
- Solución de Problemas
- Relación con otras disciplinas
- Ingeniería de Software e Ingeniería de Sistemas
- Ética y Responsabilidad
- Calidad del Software
- Participantes en el Desarrollo
- 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

Conferencia OTAN 1968 - “Crisis del Software”



Definiciones (IEEE)

- **Software:** Programas de computador, procedimientos, y la documentación y los datos posiblemente asociados relacionados con la operación de un sistema de computador - IEEE 90.
- **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:
 - Métodos (o Técnicas)
 - Herramientas (mejorar resultado o eficiencia)
 - Procedimientos (combinan métodos y herramientas)
 - Paradigmas (“estilo” de la solución)

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

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

Tipos de Software

- Productos Genéricos
 - COTS (Commercial Off The Shelf)
 - SAAS (Cloud)
 - Personalización
- Productos Personalizados (Desarrollo a Medida)
 - Cliente específico
 - SubContratos
 - Mantenimientos
- Soluciones “llave en mano”

CLIENTE



**Patrocina el desarrollo
del sistema**

\$\$\$,
necesidades

USUARIO



**Usa el
sistema**

Obligación
contractual

DESARROLLADOR



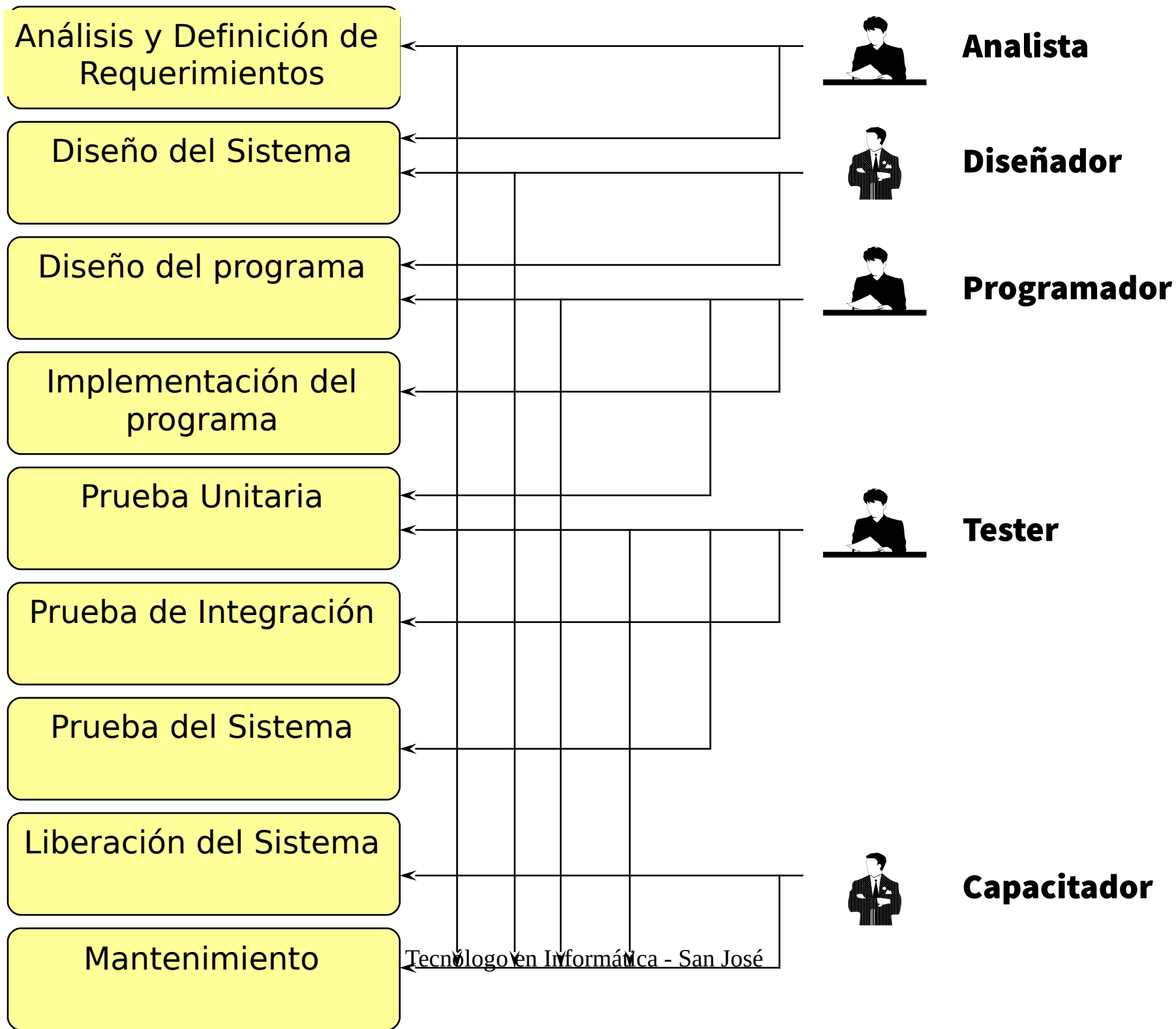
**Construye
el sistema**

Necesidades

Sistema de software

Tecnólogo en Informática - San José

Paso en el Desarrollo de Software



Roles de los Desarrolladores

Ética y Responsabilidad

- El Software está presente en casi todas las actividades humanas
- Repercusiones de fallas en el software:
 - pérdidas financieras
 - riesgo a la seguridad
 - ...
- Más allá de las fallas:
 - Impacto social
 - Calidad de vida
 - Cuestiones legales

Calidad

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

Calidad (falta-falla)



un error humano

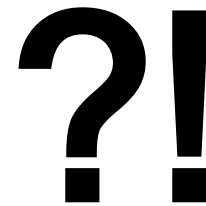
→
puede generar



**una FALTA
(interna)**



que puede generar



**una FALLA
(externa)**

Calidad

- El 30% del costo de bienes y servicios es por errores y desperdicios:

NO CALIDAD

- El 85% de los desperdicios no corresponden a errores humanos sino a sistemas de trabajo mal diseñados
- Un cliente satisfecho lo comunica a 5 personas, uno insatisfecho... a 20.
- Es cinco veces más difícil conseguir un nuevo cliente que mantener uno.
- El costo de sustituir un artículo defectuoso en la línea de producción es bastante fácil de estimar, pero el costo de un artículo defectuoso que llega a las manos de un cliente es complejo de calcular.
- ENTONCES: La calidad no es una “moda o tendencia”... es por **DINERO**, para lograr competitividad y la supervivencia de la organización

Calidad

- **Factores de Competitividad y su evolución en el tiempo:**
 - En los años 50 y 60, el precio asociado al volumen de producción
 - En los años 70, el marketing como venta de productos
 - En los años 80, la calidad del producto
 - En los años 90, el diseño del producto
- Hoy, es el **TIEMPO**. La organización con mayor capacidad de adaptación, compite con ventaja. Implica mirar al cliente, identificar lo que necesita, diseñarlo rápidamente y producirlo con calidad a un precio atractivo

Calidad - Visiones

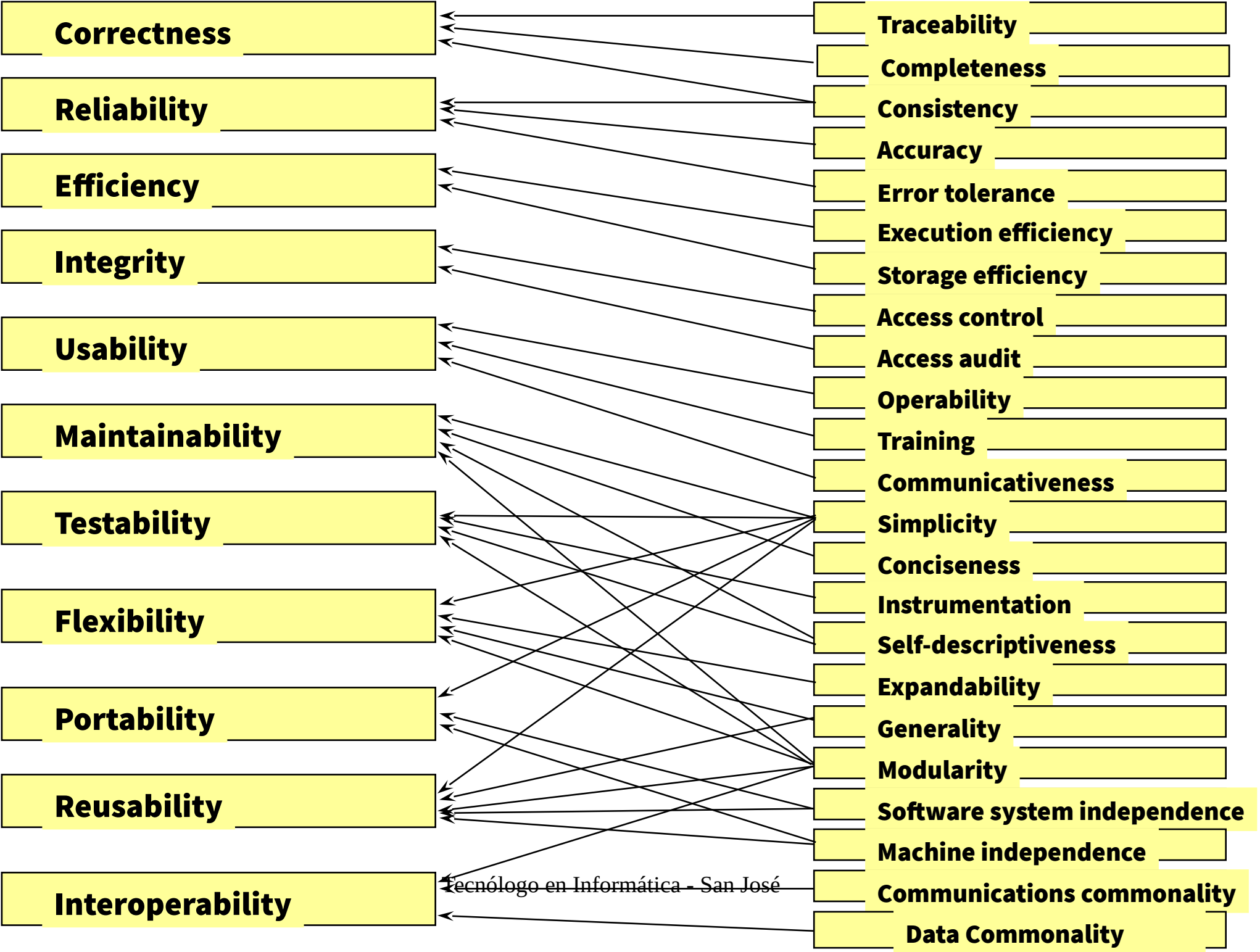
¿Qué es la Calidad?

Visión:

- 1) del Usuario - adecuación al uso
- 2) del Productor- adecuación a las especificaciones
- 3) del Producto - características específicas
 - 1) comportamiento externo (visible para todos)
 - 2) características internas (normalmente sólo visibles al productor)

Calidad – Visiones

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



Calidad - Visiones

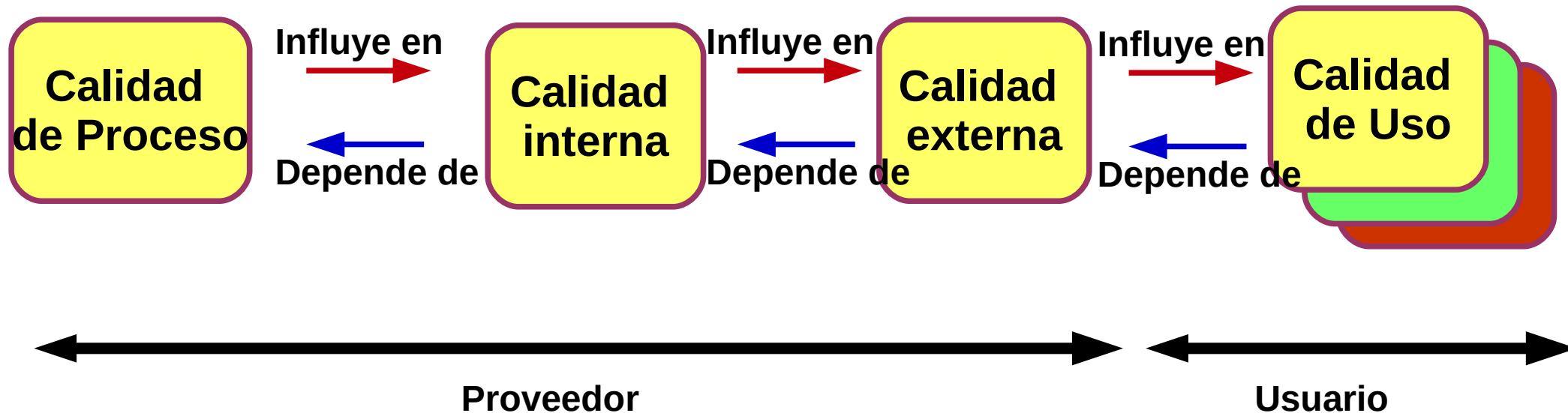
- Según la visibilidad:
 - factores externos (visibles a todos)
 - factores internos (visibles a los implementadores)
- Atinentes al:
 - Producto (una vez que el producto ya existe)
 - Proceso de Producción (mientras se produce)
- Qué relación habrá entre
 - ¿factores internos y externos?
 - ¿factores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

Calidad durante el ciclo de vida

Proceso

Producto

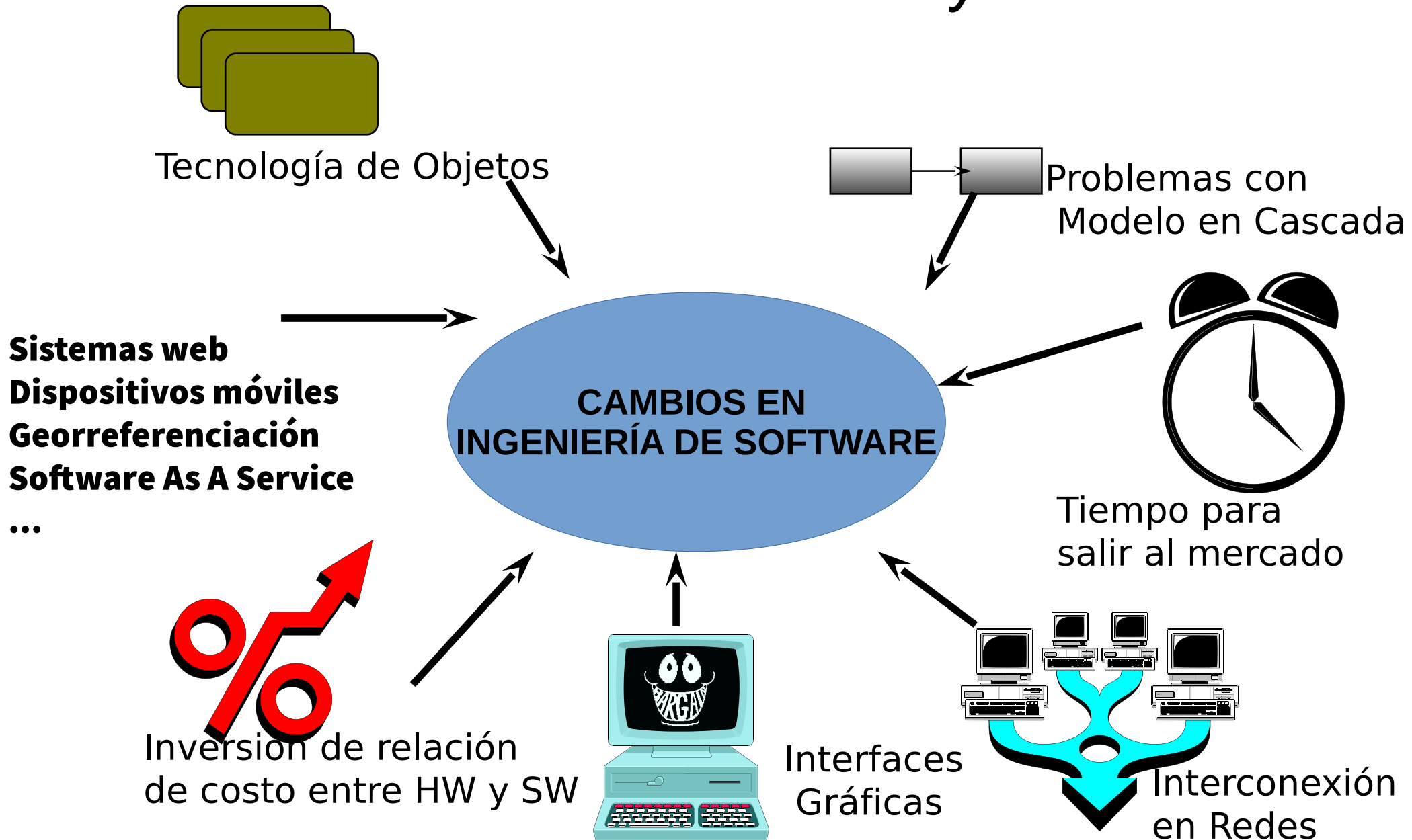
Uso



Definiciones y Preguntas Frecuentes

| Preguntas | Respuestas |
|--|--|
| ¿Qué es el software? | Son los programas de computador y la documentación asociada a los mismos. Los productos de software pueden ser desarrollados para un cliente en particular o para el mercado en general |
| ¿Cuáles son los atributos de un buen software? | Un buen software debe ofrecer la funcionalidad y el rendimiento requerido por el usuario y a su vez debe ser : mantenible, confiable y fácil de usar- |
| ¿Qué es la ingeniería de software? | La ingeniería de software es una disciplina dentro de la ingeniería que desarrolla todos los aspectos relacionados a la producción de software. |
| ¿Cuáles son las actividades fundamentales de la ingeniería de software? | Especificación, desarrollo, validación y evolución del software. |
| ¿Cuál es la diferencia entre la ingeniería de software y la ciencia de la computación? | La ciencia de la computación se focaliza en la teoría y fundamentos, mientras que la ingeniería de software tiene que ver con los aspectos prácticos de desarrollo y distribución de software- |
| ¿Cuál es la diferencia entre ingeniería de software e ingeniería de sistemas? | La ingeniería de sistemas se focaliza en todos los aspectos del desarrollo de sistemas informáticos, incluyendo hardware, software e ingeniería de procesos. La ingeniería de software es una parte de ella. |

Cambios y Tendencias



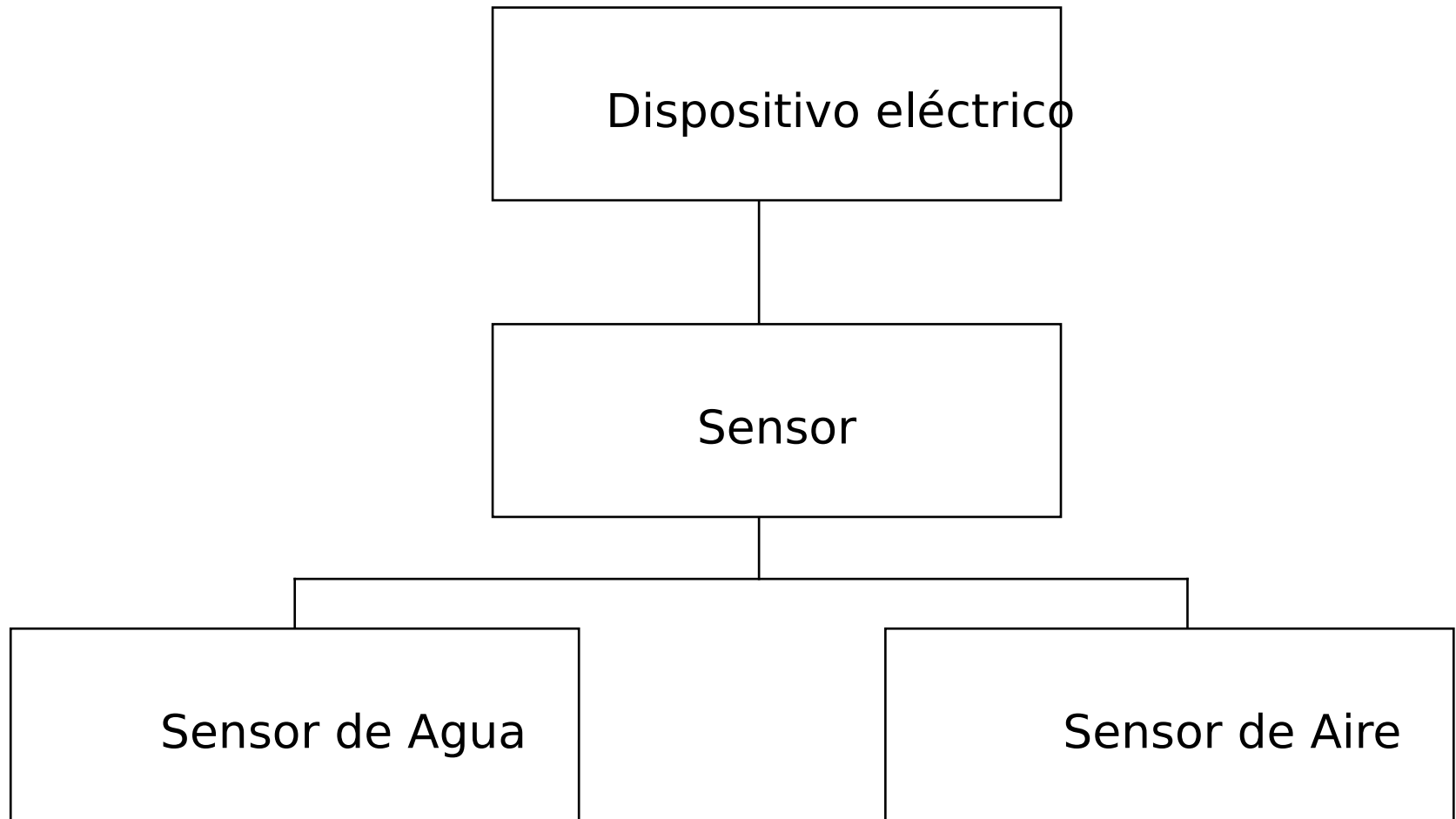
Cambios y Tendencias

- 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

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