

# Introducción al desarrollo de sistemas de información

María Mora  
Administradora del Nodo GBIF  
Costa Rica

# Temas

1. ¿Qué es un sistema de información?
2. Tipos de sistema de información.
3. Características de la información valiosa.
4. Retos en el desarrollo de sistemas.
5. Metodologías de desarrollo de sistemas
  - Desarrollo en cascada
  - Rational Unified Process (RUP)
6. Factores críticos de éxito en el desarrollo de software.

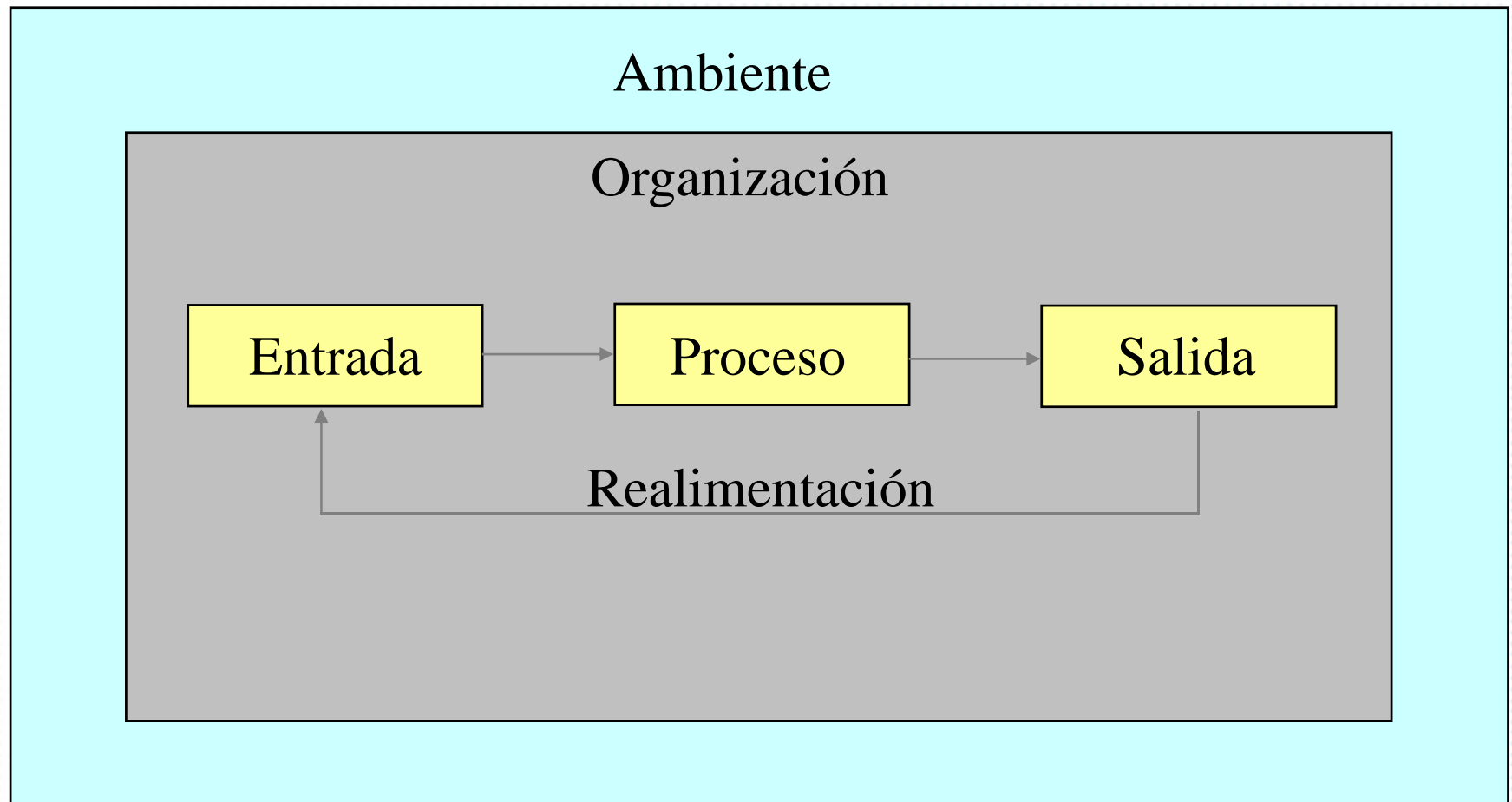
La clave para una estrategia de desarrollo en un proceso de toma de decisiones inteligentes en el uso sostenible de los recursos biológicos y el compartir equitativo de los beneficios, depende de la información sistemáticamente organizada.

*Guía para la Administración de Información bajo el contexto de la Convención sobre Diversidad Biológica,  
Jake Reynolds y John Busby del WCMC*

# 1. ¿Qué es un sistema de información?

Es un conjunto de elementos que interactúan interna y externamente, para producir información que satisface necesidades de un determinado grupo de usuarios.

# 1. ¿Qué es un sistema de información?



# 1. ¿Qué es un sistema de información?

## Aspectos clave de la definición:

1. Todo sistema es único.
2. Interactúan procedimientos, personas, hardware, software, redes de comunicaciones, otros sistemas, etc.
3. Procesos: capturar, transformar, almacenar y diseminar.
4. Existen requerimientos por cubrir.
5. Necesidad de comunicación entre las partes.

# 1. ¿Qué es un sistema de información?



## 2. Tipos de sistemas de información

- Los sistemas tradicionalmente se clasifican en: estratégicos, control y operativos.
- Por dominio de aplicación: contables, de biodiversidad, de recursos humanos, de visualización científica de datos, de administración de imágenes, etc.
- Por “nivel” de la información: basados en datos, información o conocimiento.



### 3. Características de la información valiosa

1. **Exacta** (sin errores, principio GIGO).
2. **Completa**.
3. **Económica** (balance entre costo de producirla y valor de la información).
4. **Flexible** (útil para muchos prósitos).
5. **Confiable** (calidad depende de fuente de datos y de metodología de recolección y procesamiento).

### 3. Características de la información valiosa

6. **Relevante.**
7. **Simple** (evitar sobrecarga de información).
8. **A tiempo.**
9. **Verificable** (consultas a múltiples fuentes).
10. **Accesible** (fácilmente, en el formato y el momento adecuados).
11. **Segura** (inaccesible para usuarios no autorizados)

## 4. Retos en el desarrollo de sistemas de información

1. Barreras de comunicación.
2. Intangibilidad de los productos.
3. Variabilidad de los requerimientos.
4. Dificultad para lograr estimaciones confiables.
5. Corta vida útil de productos.
6. Procesos de desarrollo no administrados como proyectos.

## 4. Retos en el desarrollo de sistemas de información

6. Disciplina relativamente nueva (ingeniería de sistemas).
7. Altas expectativas respecto de herramientas técnicas (fracasos generalmente ocurren por aspectos humanos).
8. Alto ritmo de cambio no permite asimilación.
9. Urgencia impide análisis y diseño al ritmo adecuado.

## 5. Metodologías de desarrollo de sistemas

### Desarrollo en cascada

Características del esquema tradicional:

- Proceso secuencial (en cascada).
- Productos no visibles hasta después del desarrollo.
- Orientado a la producción documental.
- Dificultad para incorporación de cambios durante el proceso.
- Genera fuertes presiones por resultados en el corto plazo.

# 5. Metodologías de desarrollo de sistemas

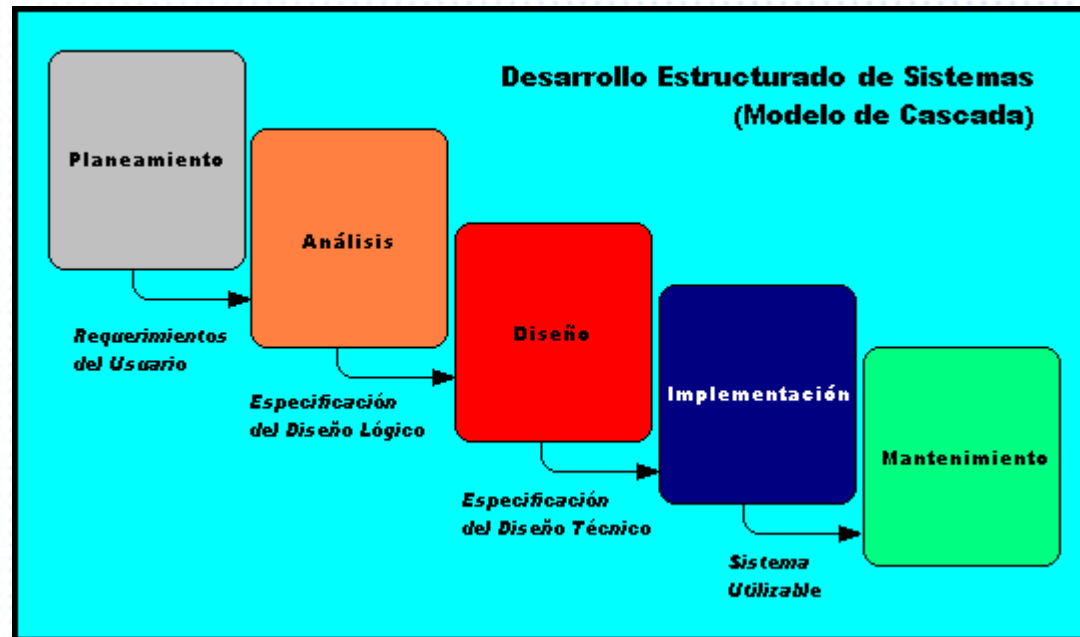
## Desarrollo en cascada

Esquema tradicional con etapas como:

- Análisis de requerimientos
- Diseño
- Implementación
- Pruebas
- Capacitación
- Instalación y paralelo
- Mantenimiento

# 5. Metodologías de desarrollo de sistemas

## Desarrollo en cascada



## 5. Metodologías de desarrollo de sistemas

### Desarrollo en cascada

Costo relativo del cambio durante las diferentes etapas:

- Análisis
  - Diseño
  - Desarrollo
  - Uso
- \$
  - \$\$
  - \$\$\$
  - \$\$\$\$



# 5. Metodologías de desarrollo de sistemas

## Desarrollo en cascada

Etapa de análisis:

- Estudiar la situación actual
- Definir los requerimientos de información.
- Recolectar datos (entrevistas, encuestas).
- Identificar productos del sistema.
- Realizar estudio de factibilidad.
- Definir especificación lógica (modelo conceptual, metodología).

# 5. Metodologías de desarrollo de sistemas

## Desarrollo en cascada

Etapas de diseño:

- Refinar modelos conceptuales.
- Diseñar modelo dinámico.
- Diseñar modelo funcional.
- Definir módulos a construir.
- Priorizar etapas de desarrollo.
- Diseñar prototipos.
- Diseñar algoritmos.
- Documentar.

# 5. Metodologías de desarrollo de sistemas

## Desarrollo en cascada

Etapa de implementación:

- Programar componentes de cada módulo.
- Probar cada componente metódicamente (programadores).
- Probar cada componente por parte de usuarios.
- Implantar los módulos.
- Documentar el sistema.

## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process (RUP)

- **Descripción general**
  - El RUP es un proceso de ingeniería de software.
  - Utiliza el paradigma de orientación a objetos para su descripción.
  - Es un *framework* de proceso configurable para satisfacer necesidades específicas.
  - Implementa las mejores prácticas de desarrollo de software.

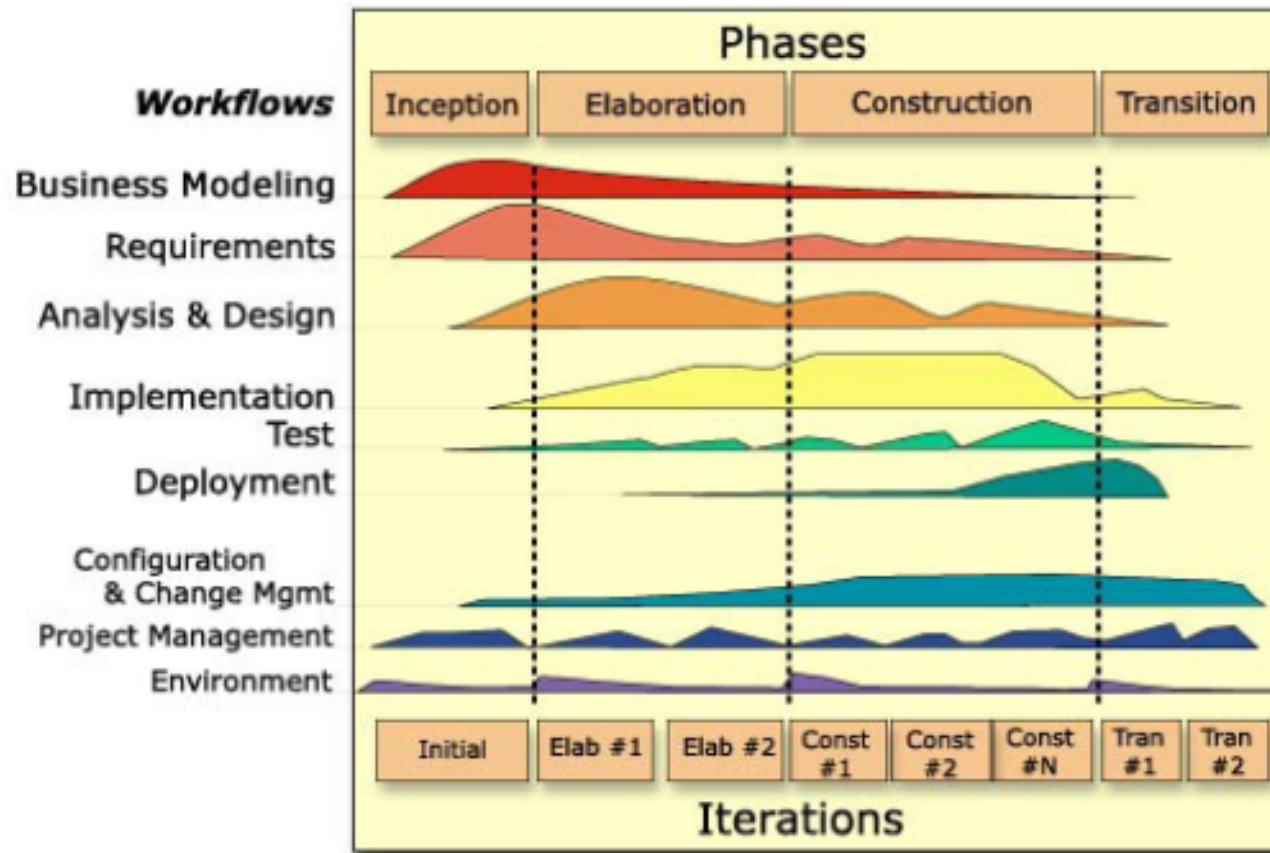
## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

- **Tres características principales**
  - Dirigido por casos de uso
    - Los casos de uso capturan requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario.
  - Centrado en una arquitectura
    - Comprende los aspectos estáticos y dinámicos más importantes del sistema.
  - Iterativo e incremental
    - El trabajo se divide en piezas pequeñas o miniproyectos; cada uno proveyendo un subproducto incremental.

# 5. Metodologías de desarrollo de sistemas

## Rational Unified Process



## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### ¿Qué es un requerimiento?

- Es una condición de capacidad que un sistema debe satisfacer.
- Se pueden agrupar en categorías según el modelo:
  - Funcionalidad
  - Usabilidad
  - Confiabilidad
  - Rendimiento
  - Apoyo

## 5. Metodologías de desarrollo de sistemas

# Rational Unified Process

### ¿Qué es un requerimiento?

- Requerimientos de funcionalidad
  - Conjunto de características
  - Capacidades
  - Seguridad
- Requerimientos de usabilidad
  - Estética
  - Consistencia en la interfaz de usuario
  - Ayuda en línea y sensitiva al contexto
  - Documentación de usuarios
  - Materiales de capacitación



## 5. Metodologías de desarrollo de sistemas

# Rational Unified Process

### ¿Qué es un requerimiento?

- Requerimientos de confiabilidad
  - Frecuencia y severidad de las fallas
  - Recuperabilidad
  - Predictibilidad
  - Precisión
- Requerimientos de rendimiento
  - Velocidad
  - Eficiencia
  - Disponibilidad
  - Productividad de procesamiento
  - Tiempo de recuperación
  - Uso de los recursos

## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### ¿Qué es un requerimiento?

- Requerimientos de apoyo
  - Habilidad de correr pruebas
  - Extensibilidad
  - Adaptabilidad
  - Mantenibilidad
  - Compatibilidad
- Requerimientos de diseño
  - También llamado restricciones de diseño

## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### ¿Qué es un requerimiento?

- Requerimientos de implementación
  - Estándares
  - Lenguajes de programación
  - Políticas de integridad de la base de datos
  - Ambientes de operación
- Requerimientos de interfaz
  - Elementos externos con los que el sistema debe interactuar.

## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### Casos de uso

- Secuencia de pasos que un sistema realiza para proveer un resultado de valor para un actor particular.
- Se enfocan en la funcionalidad del sistema y necesitan de requerimientos adicionales para proveer una especificación completa de los requerimientos de software.
- Se concentra en el **qué** y **no en el cómo**.

## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### Casos de uso

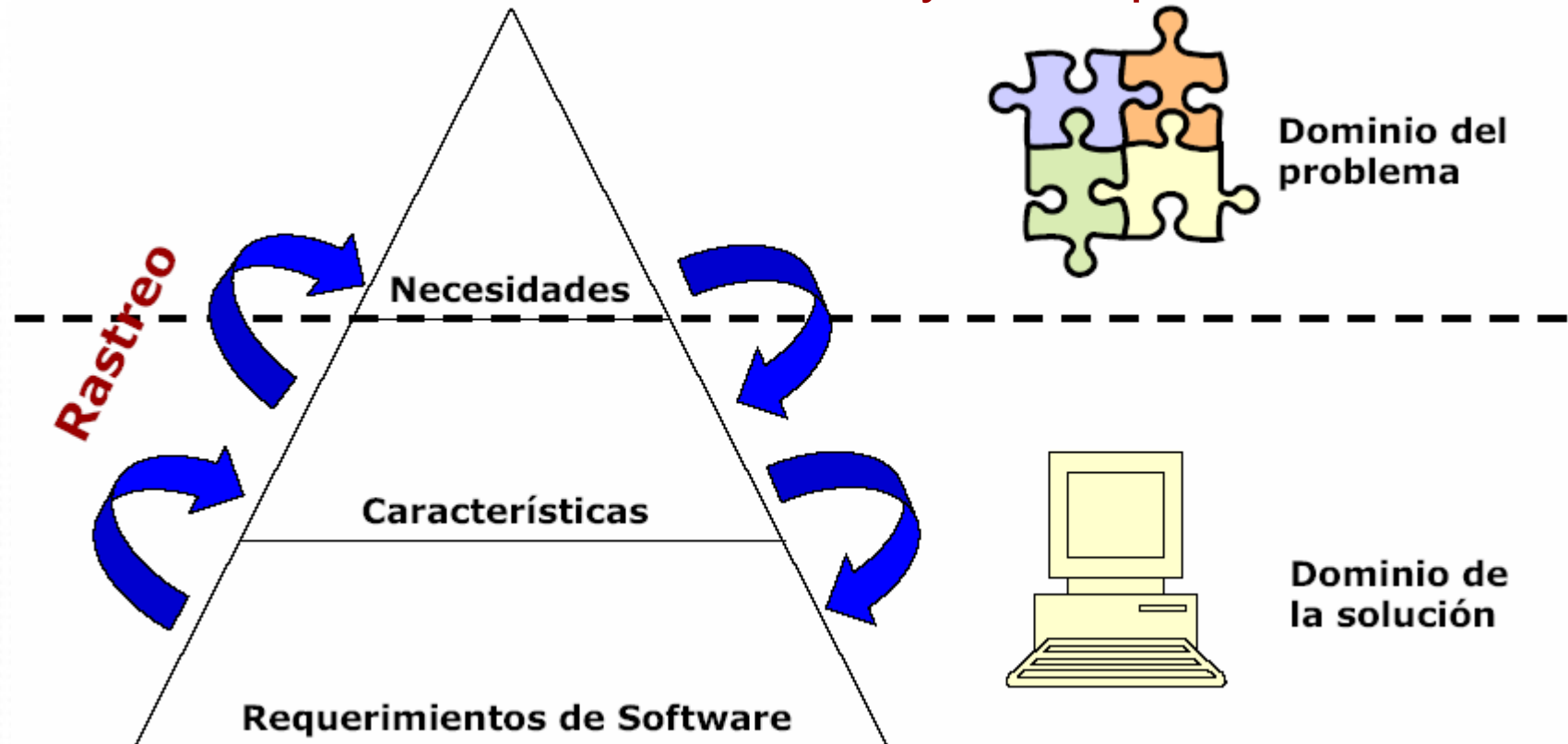
- Los casos de uso se utilizan principalmente para capturar los requerimientos de comportamiento de un sistema.
- Posicionan los requerimientos de software en contexto: muestran cómo el sistema provee valor a sus patrocinadores al mismo tiempo que los hace más fáciles de entender.

# 5. Metodologías de desarrollo de sistemas

## Rational Unified Process

### Casos de uso

- Relación entre los casos de uso y los requerimientos



## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### Casos de uso

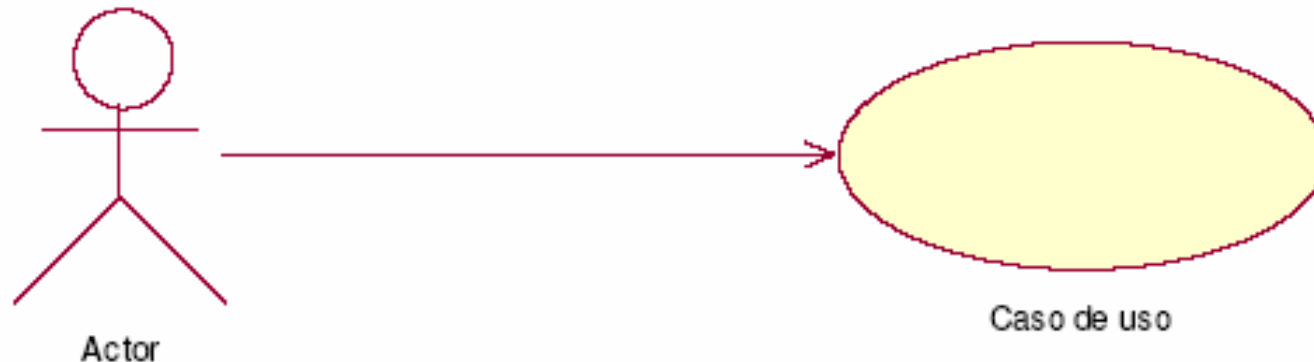
- Principios generales de los casos de uso
  - No existen en aislamiento
    - Se deben considerar aspectos económicos, tecnológicos, políticos y del negocio y cómo el sistema afectará esos aspectos.
  - Son una herramienta sintética mas que analítica.
    - El problema es que nos ahogamos en un mar de requerimientos.

# 5. Metodologías de desarrollo de sistemas

## Rational Unified Process

### Casos de uso

- Elementos básicos de un caso de uso





## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### Arquitectura de software

- ¿Qué es arquitectura de software?

El conjunto de decisiones significativas acerca de la organización de un sistema, la selección de los elementos estructurales, sus interfases y comportamiento y la composición de estos elementos en subsistemas progresivamente más grandes.

## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### Arquitectura de software

- ¿Qué es arquitectura de software?

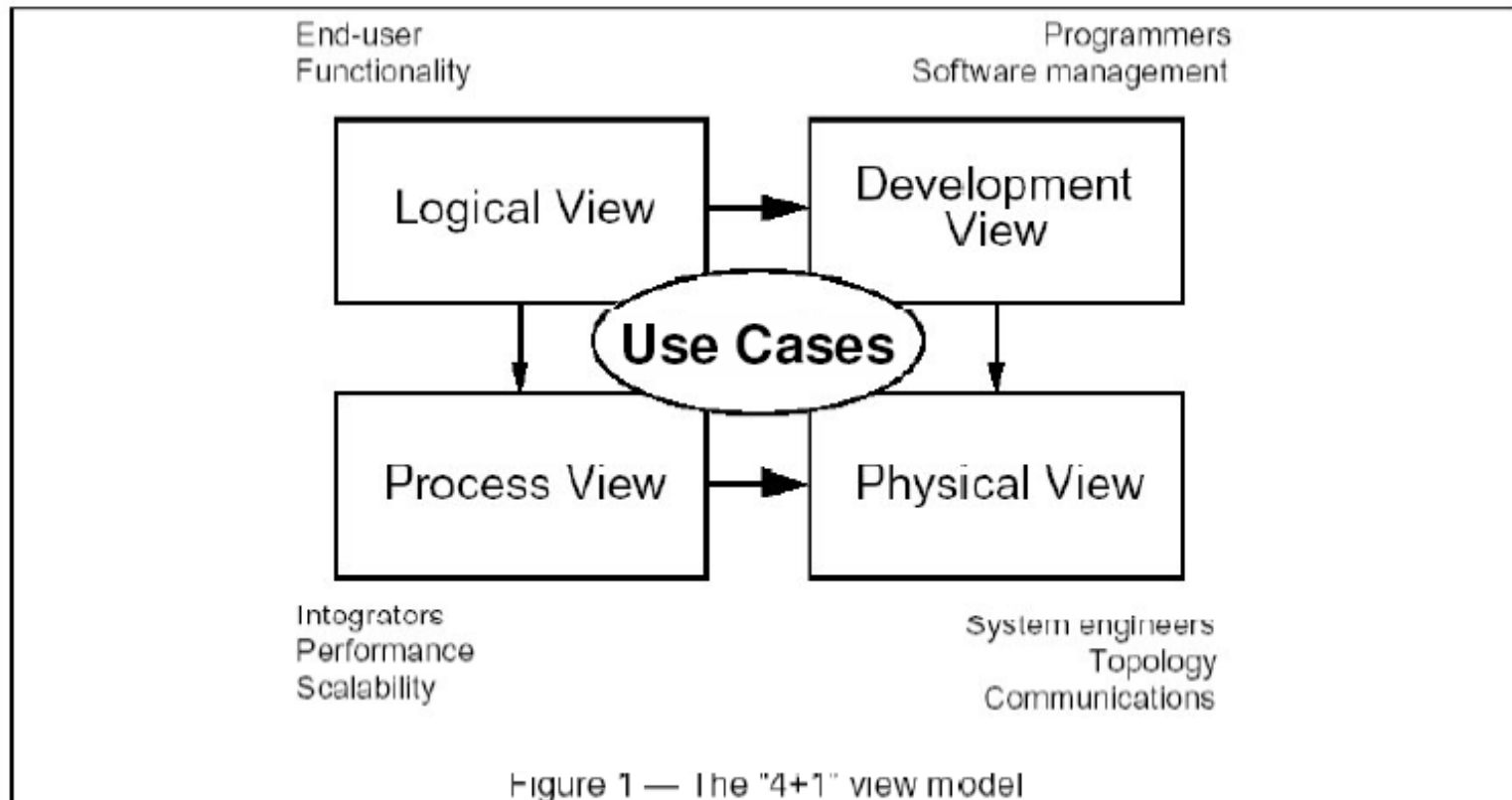
La arquitectura de software no se limita solamente a la estructura y el comportamiento: incluye usabilidad, rendimiento, robustez, reutilización, restricciones económicas y tecnológicas, además de preocupaciones sobre la estética.

# 5. Metodologías de desarrollo de sistemas

## Rational Unified Process

### Arquitectura de software

- Las 4 + 1 vistas



## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

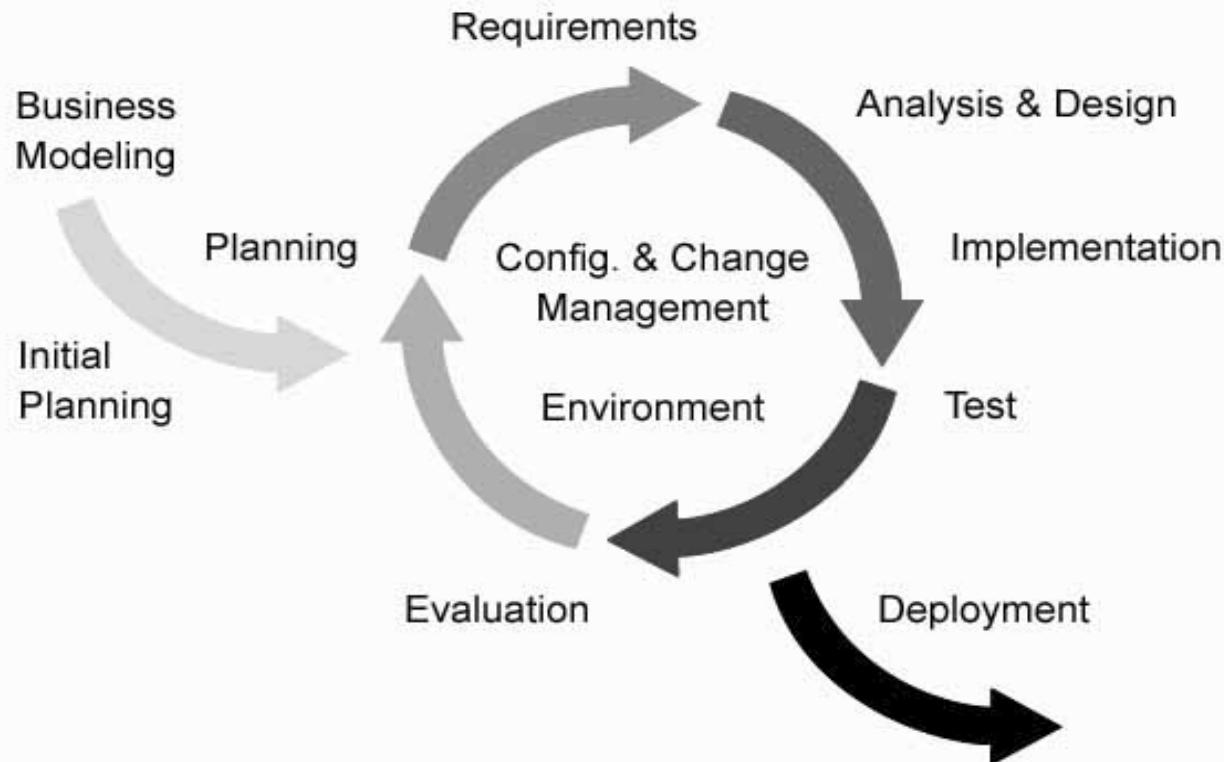
#### Proceso iterativo e incremental

- Proceso de construir sistemas haciendo aproximaciones que se acercan progresivamente a la solución ideal.
- Se obliga a identificar los riesgos del proyecto en etapas tempranas.
- Es un enfoque de descubrimiento, invención e implementación continuos.
- Cada iteración termina en una forma predecible y repetible.

# 5. Metodologías de desarrollo de sistemas

## Rational Unified Process

### Proceso iterativo e incremental



## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### Proceso iterativo e incremental

- **Ventajas**

- Reducción de riesgos basado en la retroalimentación temprana.
- Pruebas continuas e iterativas promueven una mejor evaluación del estado del proyecto.
- Los patrocinadores reciben evidencia concreta del avance del proyecto.
- Se pueden acomodar mejor los cambios (requerimientos, tácticos y tecnológicos).
- Los problemas más complejos se atacan primero.

## 5. Metodologías de desarrollo de sistemas

### Rational Unified Process

#### UML

- Lenguaje unificado de modelación.
- Es la notación estándar para visualizar, especificar, construir y documentar artefactos de un sistema.
- Combina las mejores técnicas de:
  - Modelación de datos (ERD)
  - Modelación del negocio (flujos de trabajo)
  - Modelación de objetos
  - Modelación de componentes.

## 6. Factores críticos de éxito en el desarrollo

- No enfrascarse en la solución perfecta.
- Diferenciar entre medios y fines.
- Establecer canales de comunicación sólidos.
- Establecer un horizonte de largo plazo en los requerimientos, pero de oportunidad en su implementación.
- Administración formal del proceso.



## 6. Factores críticos de éxito en el desarrollo

- Alta participación de los actores involucrados en el proceso (equipo de trabajo).
- Evitar indispensabilidad (de programadores y usuarios).
- Identificar y divulgar indicadores de avance del proceso.

## **Descripción de la metodología RUP utilizada en la charla preparada por:**

Herbert Barrientos

Arquitecto de software

CTI Arcobaleno

San José, Costa Rica

(506) 234-6643

hbarr@acm.org

# Preguntas

¡Muchas Gracias!