

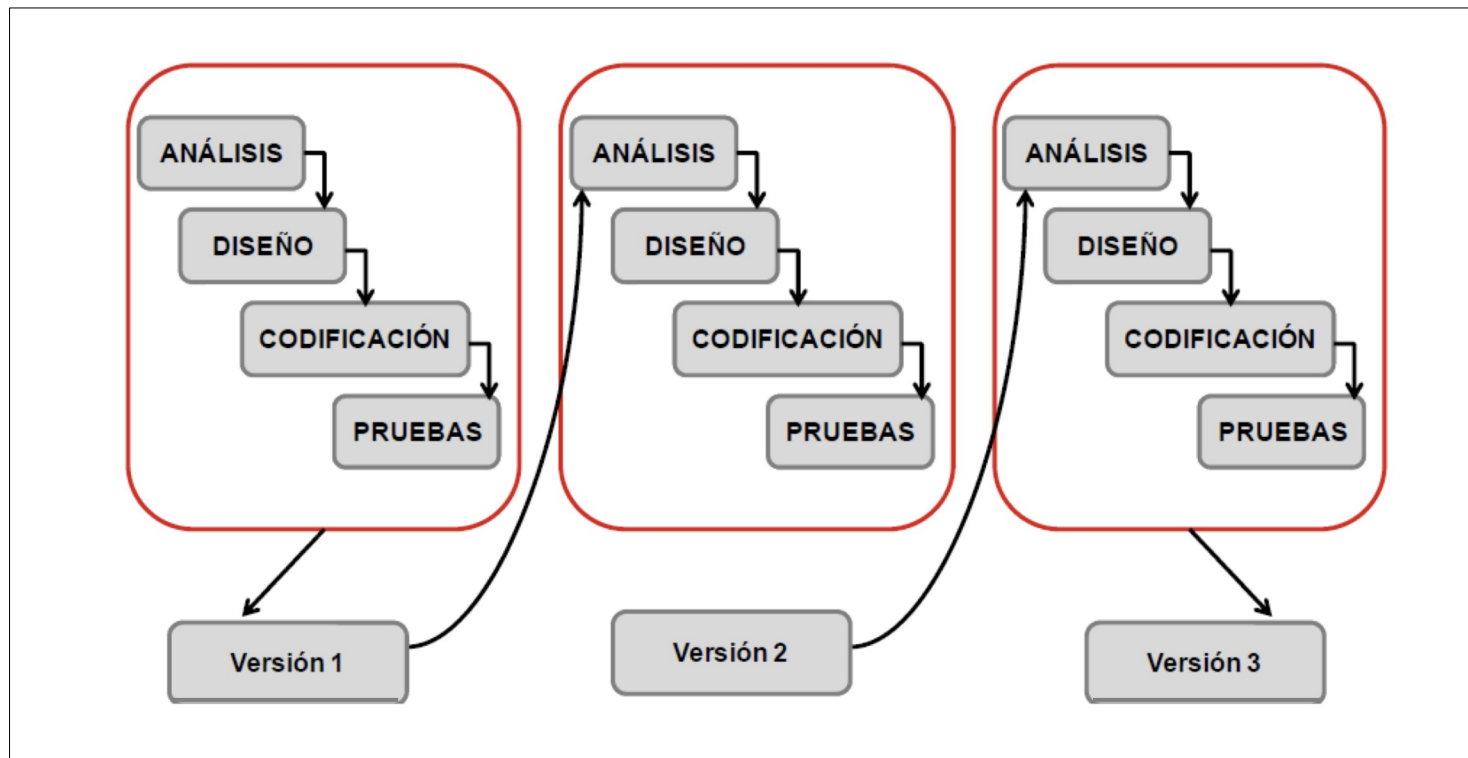
# Ingeniería de Software

## Introducción

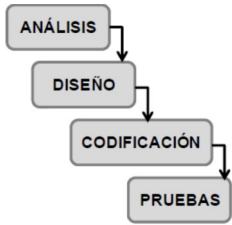
Sommerville capítulo 1

# Proceso de Software

Cuando se describe un proceso, generalmente hablamos de las actividades de dicho proceso y el orden en el cual se deberían realizar



coronaTickets.uy: Proceso Iterativo e Incremental



# Análisis

---

Objetivo: Comprender mejor la realidad y el problema

Modelado conceptual (una de las actividades):

- Realizar modelos resulta clave para el análisis de requisitos
- Ayudan a entender la situación en la cual ocurren los problemas o necesidades, así como representar soluciones
- Modelado de Dominio
  - Diagrama de Clases
- Especificación del Comportamiento
  - Diagramas de Secuencia del Sistema y Contratos

# Análisis

---

¿Qué actividades agregarían si tuvieran que interactuar con Stakeholders (interesados)?

Para obtener información acerca del dominio de la aplicación, los servicios que debe proporcionar el sistema, los requisitos de performance del sistema, restricciones de hardware, otros sistemas...

# De Análisis a Ingeniería de Requisitos

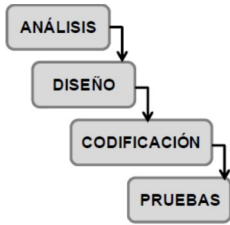
---

Es el proceso de establecer qué servicios (o funcionalidades) son requeridos y las restricciones acerca de la operación y el desarrollo

## Proceso de Ingeniería de Requisitos

- Relevamiento y **análisis** de requisitos: ¿Qué es lo que los Stakeholders requieren y esperan del sistema?
- Especificación de requisitos: definir los requisitos en detalle
- Validación de requisitos: comprobar la validez de los requisitos
- Gestión de requisitos

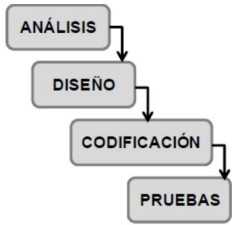
# Diseño



Consiste en dos actividades que se encuentran entre el análisis de requisitos del software y la construcción del software:

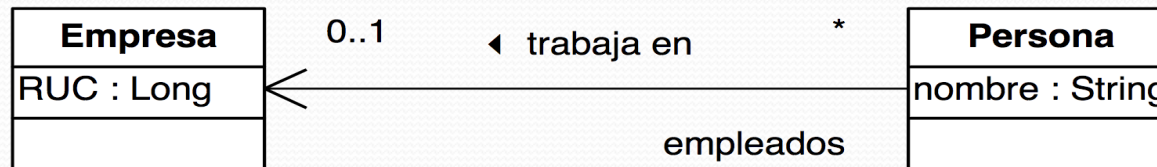
- Diseño arquitectónico de software (a veces llamado diseño de alto nivel): desarrolla la estructura y la organización de alto nivel del software e identifica los diversos componentes
- Diseño detallado del software: especifica cada componente con detalles suficientes para facilitar su construcción

Se define como "el proceso de definición de la arquitectura, componentes, interfaces y otras características de un sistema o componente" y "el resultado de [ese] proceso"



# Codificación

Consiste en transformar en código fuente, en el lenguaje de programación escogido, las construcciones definidas en el diseño



Ejemplo en C++

```
class Persona {  
    private:  
        String nombre;  
        Empresa * empresa; // pseudoatributo  
    public:  
        ...  
};
```

# De Codificación a Construcción

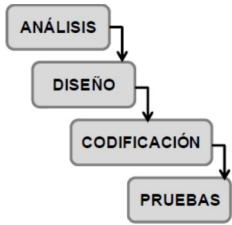
---

Refiere a la creación detallada del software mediante una combinación de **codificación**, verificación, pruebas unitarias, pruebas de integración y debugging

Implica:

- Elección del método de construcción
- El orden en el cual se crean los componentes, se integran, se gestiona la calidad de la construcción y la asignación a los distintos programadores





# Pruebas

---

Proceso de ejecutar software con el fin de provocar fallas

Niveles de prueba

- De módulo, componente o unitaria
- De integración
- De sistema
- De aceptación

Técnicas de prueba

- Estática
- Dinámica

# De Pruebas a Verificación y Validación

---

Verificación: ¿estamos construyendo el producto correctamente?

- Busca comprobar que el sistema cumple con los requisitos especificados (funcionales y no funcionales)

Validación: ¿estamos construyendo el producto correcto?

- Busca comprobar que el software hace lo que el usuario espera

Requiere llevar una planificación cuidadosa para obtener el máximo provecho de las revisiones y las **pruebas** para controlar los costos del proceso de V&V

# Liberación de Software

---

Comprende todas las actividades que son necesarias, luego de que se ha construido y probado, para comenzar el uso de un software

# Evolución de Software

---

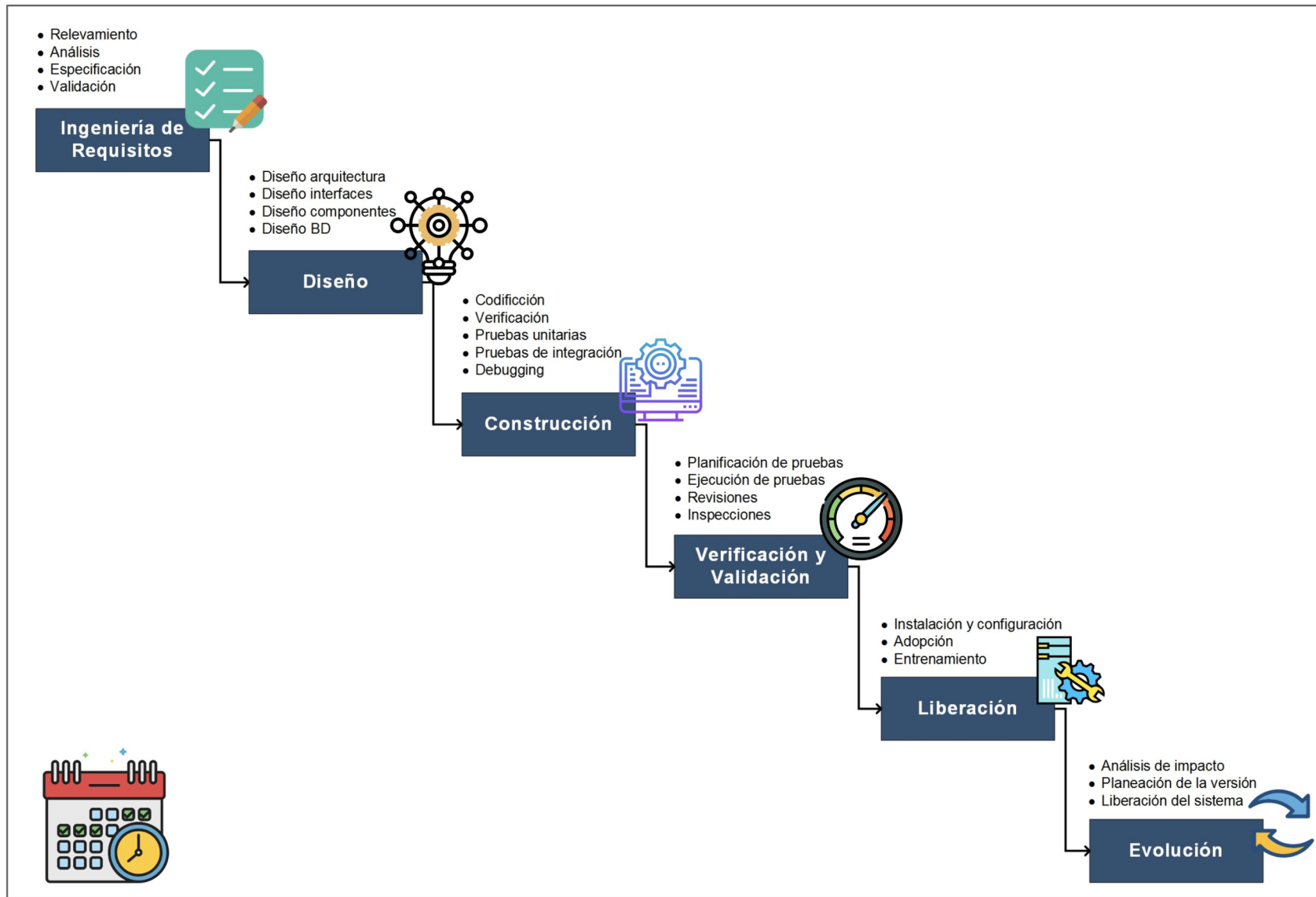
Importancia: Para mantener el valor, los sistemas deben ser modificados y actualizados

Las propuestas de cambio guían la evolución del software:  
¿Qué cambios se les ocurre?

Algunas actividades:

- Análisis de impacto
- Planeación de la versión
- Liberación del sistema

# En resumen



Proceso en casada

# Gestión de Proyectos

---



“La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo” -PMBOK 5° edición-

Objetivos principales de la gestión de proyectos:

- Entregar el software en tiempo acordado
- Mantener los costos dentro del presupuesto acordado
- Entregar un software que cumpla con las expectativas del cliente
- Mantener un equipo funcionando

# Ingeniería de Software

---

Es una disciplina de la ingeniería que se ocupa de todos los aspectos de la producción de software desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema una vez que se ha puesto en uso

- Disciplina de ingeniería

Usar teorías y métodos teniendo en cuenta las limitaciones organizacionales y financieras

- Todos los aspectos de la producción de software

También gestión de proyectos y desarrollo de herramientas, métodos, etc

→ Involucra compromisos