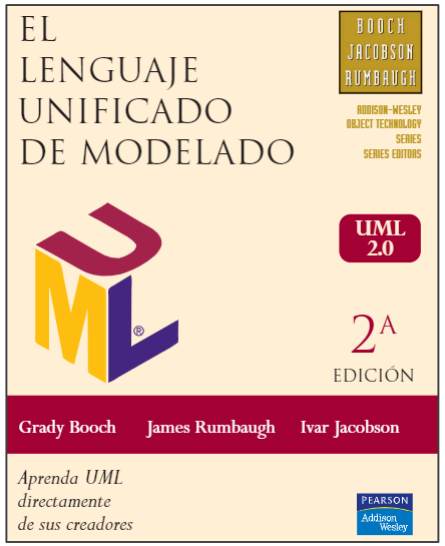
Programación Orientada a Objetos I

UML – INTRODUCCIÓN

Bibliografía

Bibliografía UML para la materia:

El Lenguaje Unificado de Modelado: Guía de usuario (2ª Edición)



Grady BOOCH – James RUMBAUGH – Ivar JACOBSON

Editorial Pearson – Addison Wesley

Disponible en biblioteca de la UCA (en forma online)

Que es UML

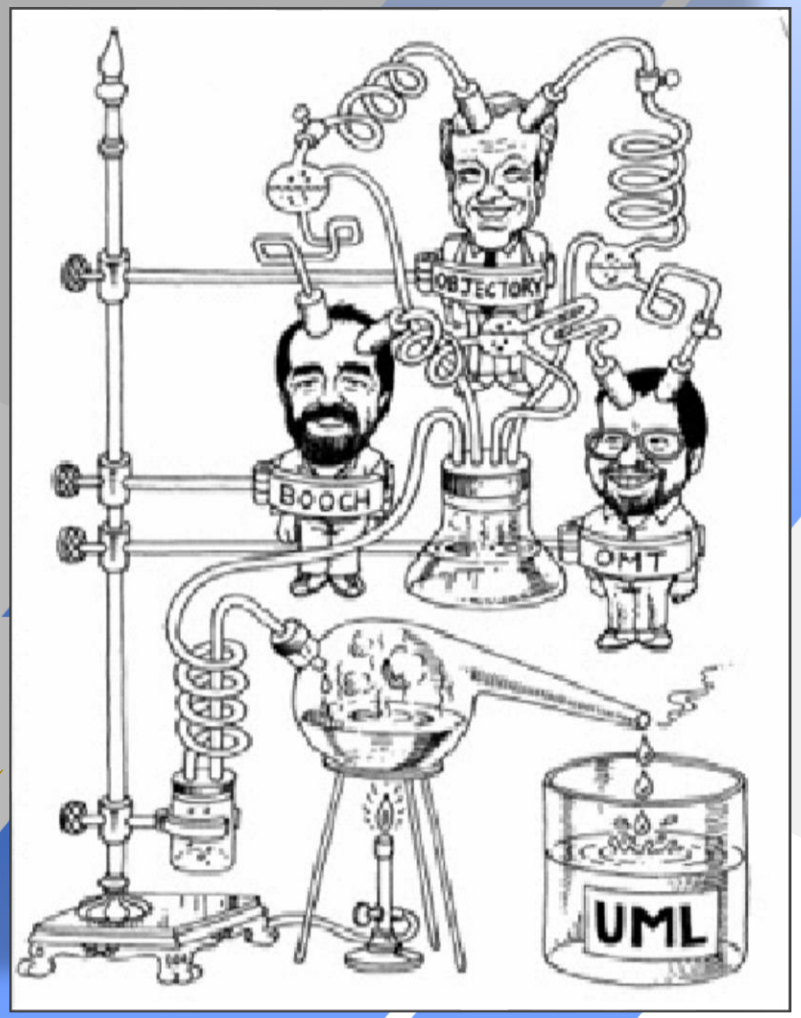
UML (Unified Modeling Language): Es un lenguaje gráfico que permite modelar, construir y documentar los elementos que constituyen un sistema software orientado a objetos. Ofrece un conjunto de modelos estándar utilizados para el diseño de proyectos.

Creado por: Grady Booch, James Rumbaugh e Ivar Jacobson

- Es un lenguaje (o una familia de lenguajes)
- No es un proceso
- No es una metodología

  Modelo = Representación abstracta de la realidad.

- UML no describe la implementación de esos modelos.



Por qué modelar

- Un modelo es la representación de algo que existe en el mundo real. (Existencia física o potencial).
- Proporciona los planos de un sistema, con mayor o menor detalle de acuerdo a lo necesario
- Incluyen aquellos elementos que tienen una gran influencia, y omite aquellos que no son relevantes
- Hay sistemas complejos y no podemos comprenderlo en su totalidad
- Es una forma de aprender más del dominio
- Podemos describir desde diferentes perspectivas (diferentes modelos)
- También como forma más económica de estudiar algo del mundo real

Un modelo es una simplificación de la realidad

Construimos modelos para comprender mejor el sistema que estamos desarrollando

Modelado

Objetivos:

Modelado

Visualizar:

- Poder ver y entender cómo es o como queremos que sea un sistema
- En forma gráfica, más fácil de entender y visualizar
- Bien definida a través de símbolos, con una semántica definida (interpretación)
- No se puede ver todo el código, pero se pueden ver y entender los diagramas
- Trasciende las fronteras de las organizaciones
- Todos lo entienden

Modelado

Especificar:

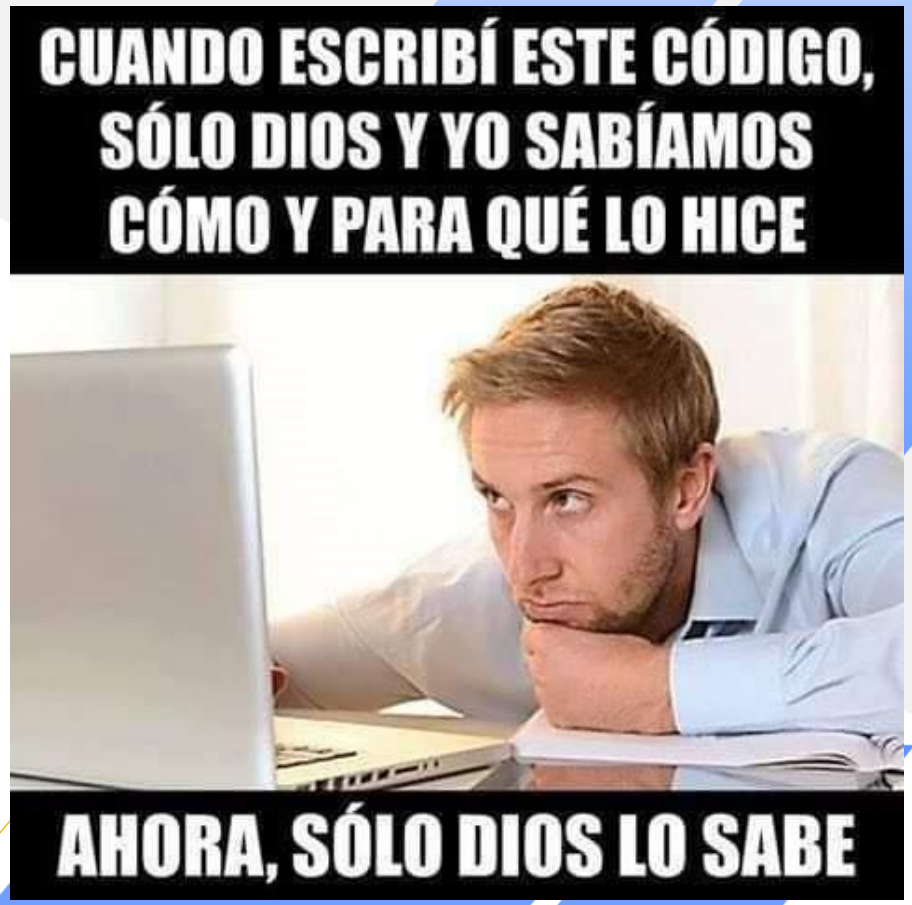
- Construir modelos precisos, no ambiguo y completos.
- Cubrir las especificaciones en las etapas de:
  - Análisis
  - Diseño
  - Implementación

Modelado

Construir:

Los modelos nos proporcionan plantillas que nos guían en la construcción de un sistema

- Se puede conectar fácilmente con lenguajes de programación
- Se puede conectar fácilmente con bases de datos
- Permite ingeniería directa
- Permite ingeniería inversa



Modelado

Documentar:

- Requisitos
- Arquitectura
- Diseño
- Código fuente
- Planificación de proyectos
- Pruebas
- Prototipos
- Versiones

Limitaciones

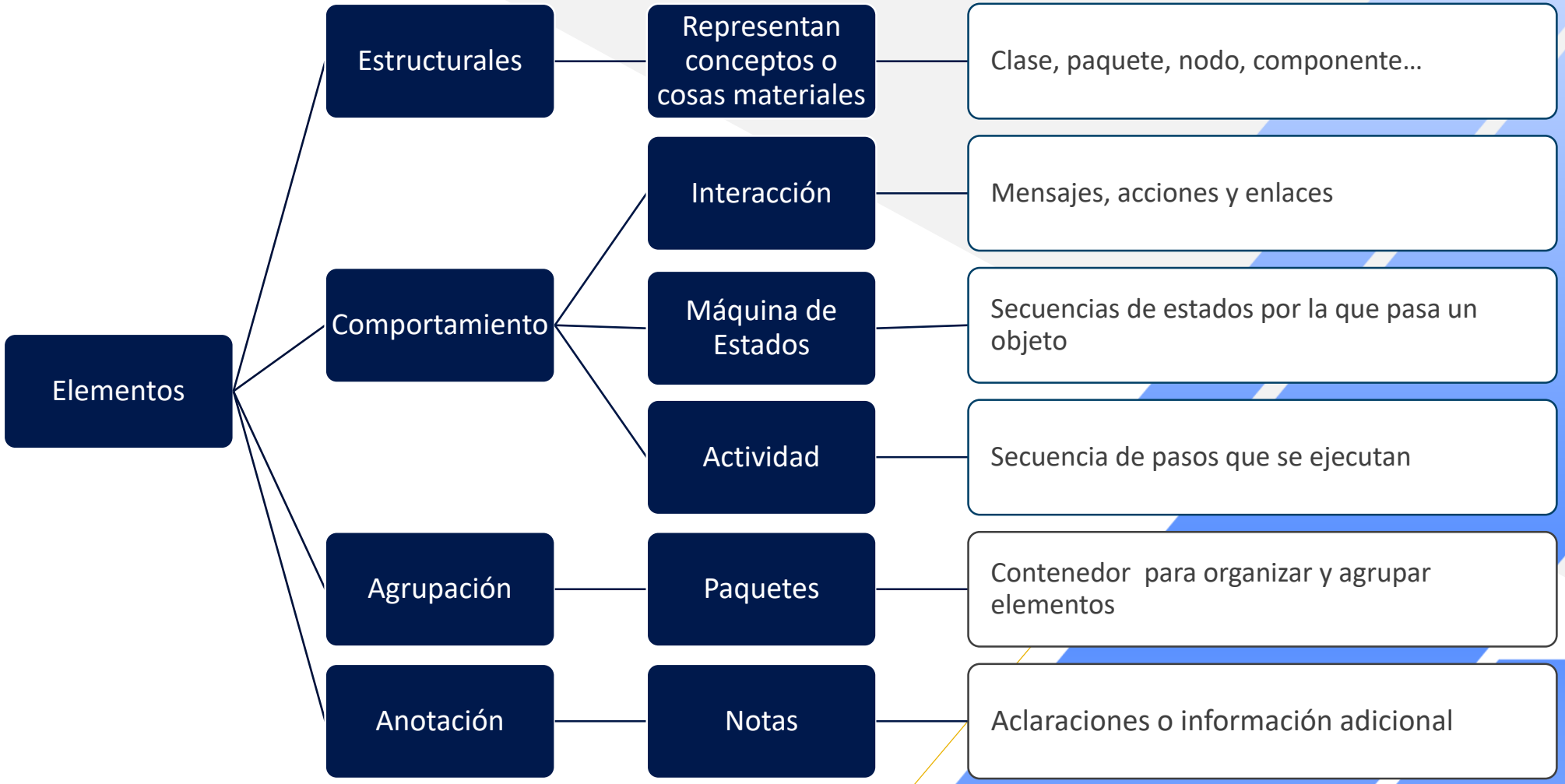
- UML ***no es un modelo de proceso*** de desarrollo de software.
- UML ***no define un ciclo de vida***, ni objetivos, ni actividades.
- UML ***no es una metodología***.
- UML ***no prescribe buenas prácticas***.
- UML ***no implementa estrategias*** para alcanzar objetivos.

UML fue concebido para describir modelos de software

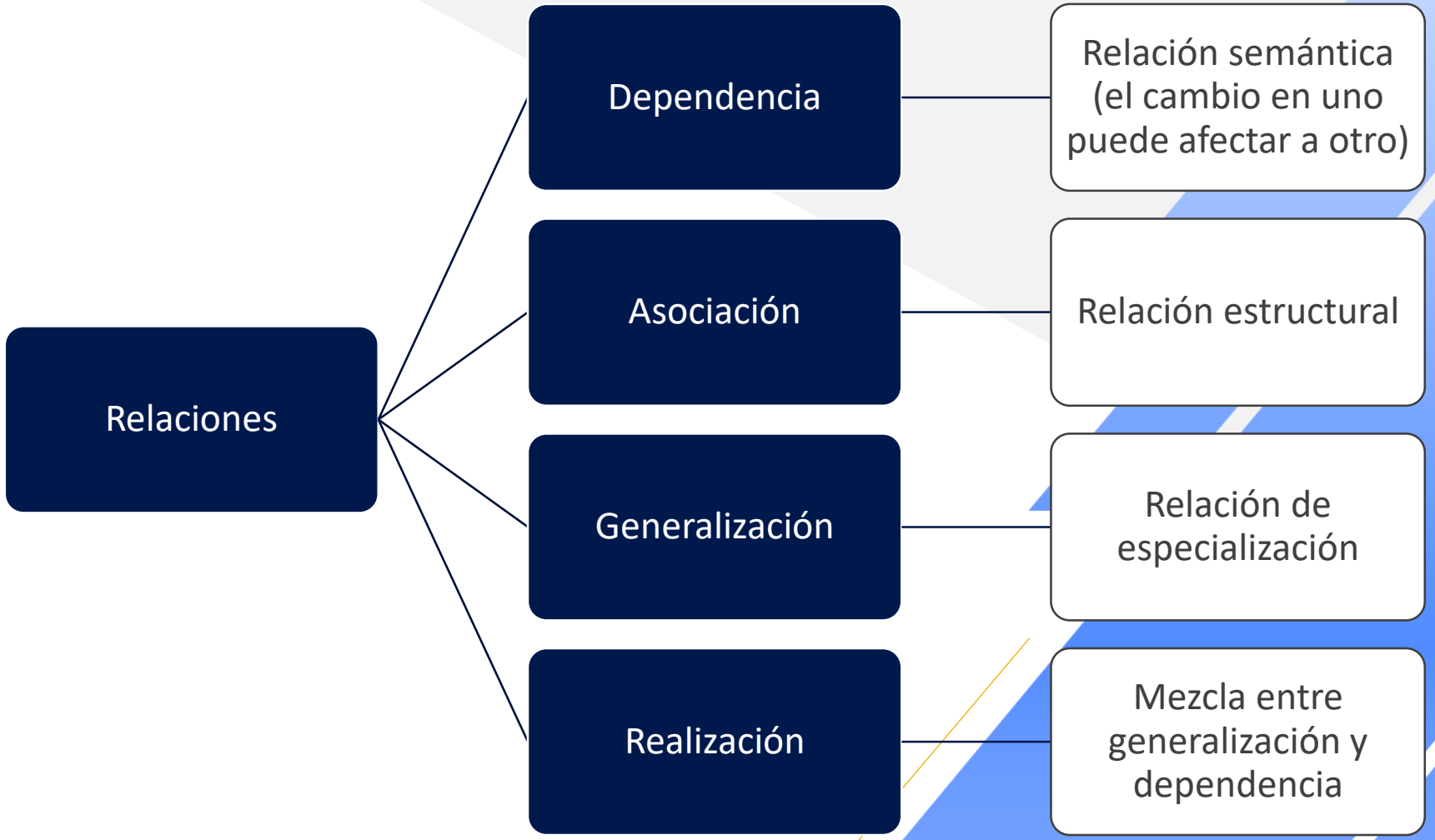
UML está fuertemente vinculado a la OO

UNIFIED
MODELING
LANGUAGE

Bloques básicos



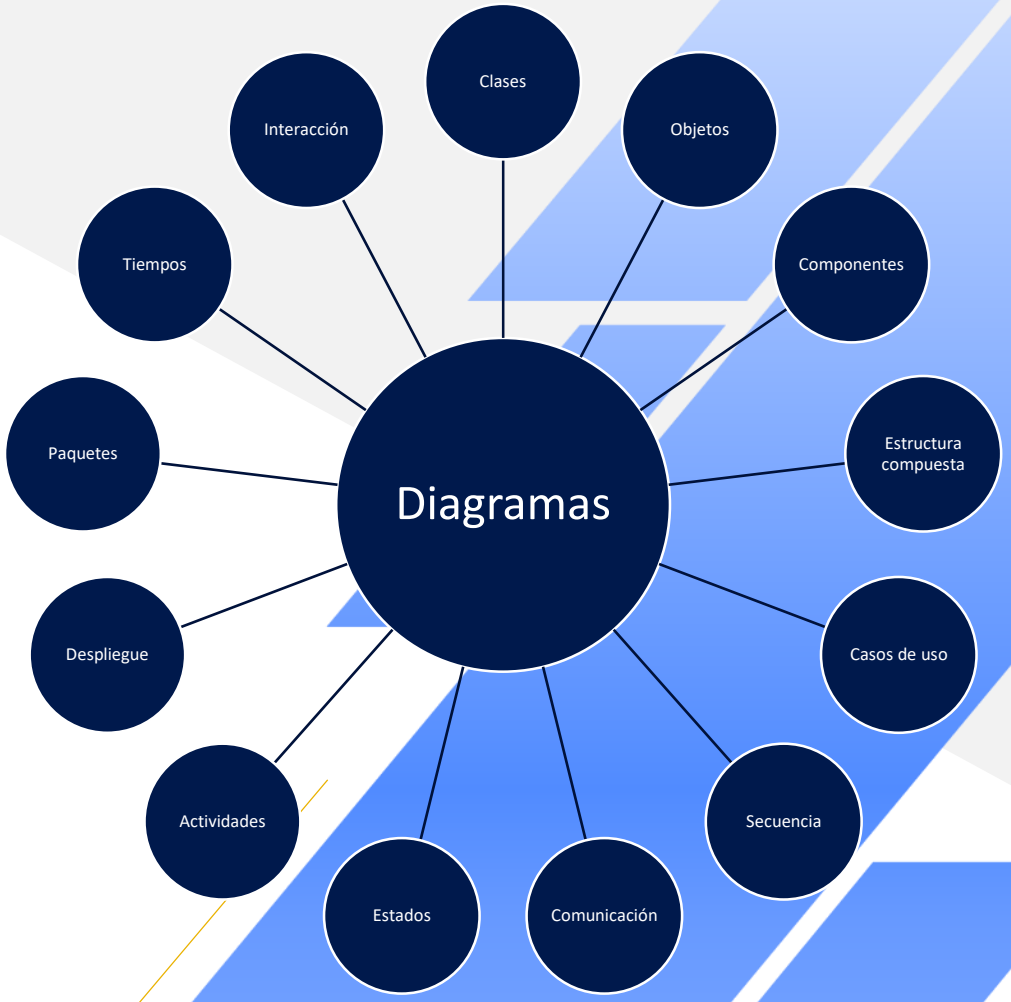
Bloques básicos



Bloques básicos

Diagramas

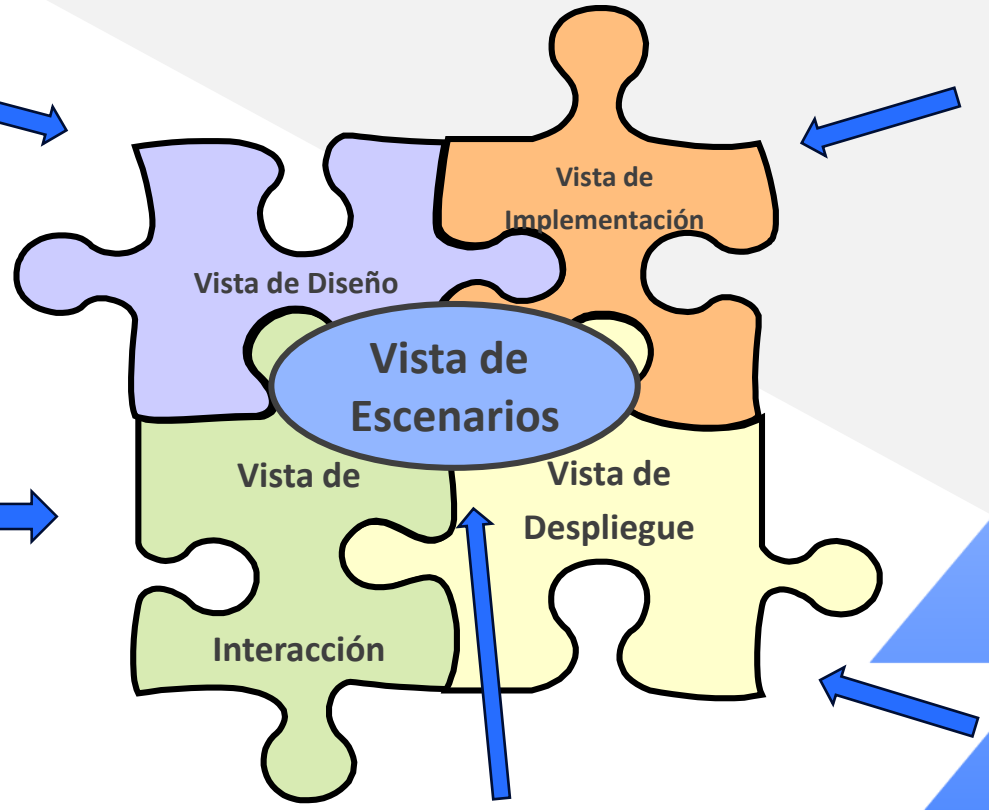
- ***Diagrama de Clases***
- ***Diagrama de Casos de Uso***
- ***Diagrama de Estados***
- ***Diagrama de Actividad***
- Diagrama de Secuencia
- Diagrama de Colaboración
- Diagrama de Componentes
- Diagrama de Paquetes
- Diagrama de Despliegue



Arquitectura: Vistas 4 + 1

Vista de Diseño:
Funcionalidad a los ***usuarios finales.*** Lo que el sistema ***debe hacer***, los ***requerimientos funcionales***.

Vista de Interacción:
representa el flujo de control entre las partes del sistema, incluyendo la concurrencia y sincronización, enfocándose en el rendimiento, escalabilidad y capacidad de procesamiento



Vista de Implementación:
abarca los artefactos para ensamblar y desplegar el sistema físico, gestionando configuraciones de versiones y la correspondencia entre clases/componentes lógicos y artefactos físicos

Vista de Despliegue: representa los nodos que conforman la infraestructura hardware donde se ejecuta el sistema, centrándose en la distribución, entrega e instalación de sus componentes físicos.

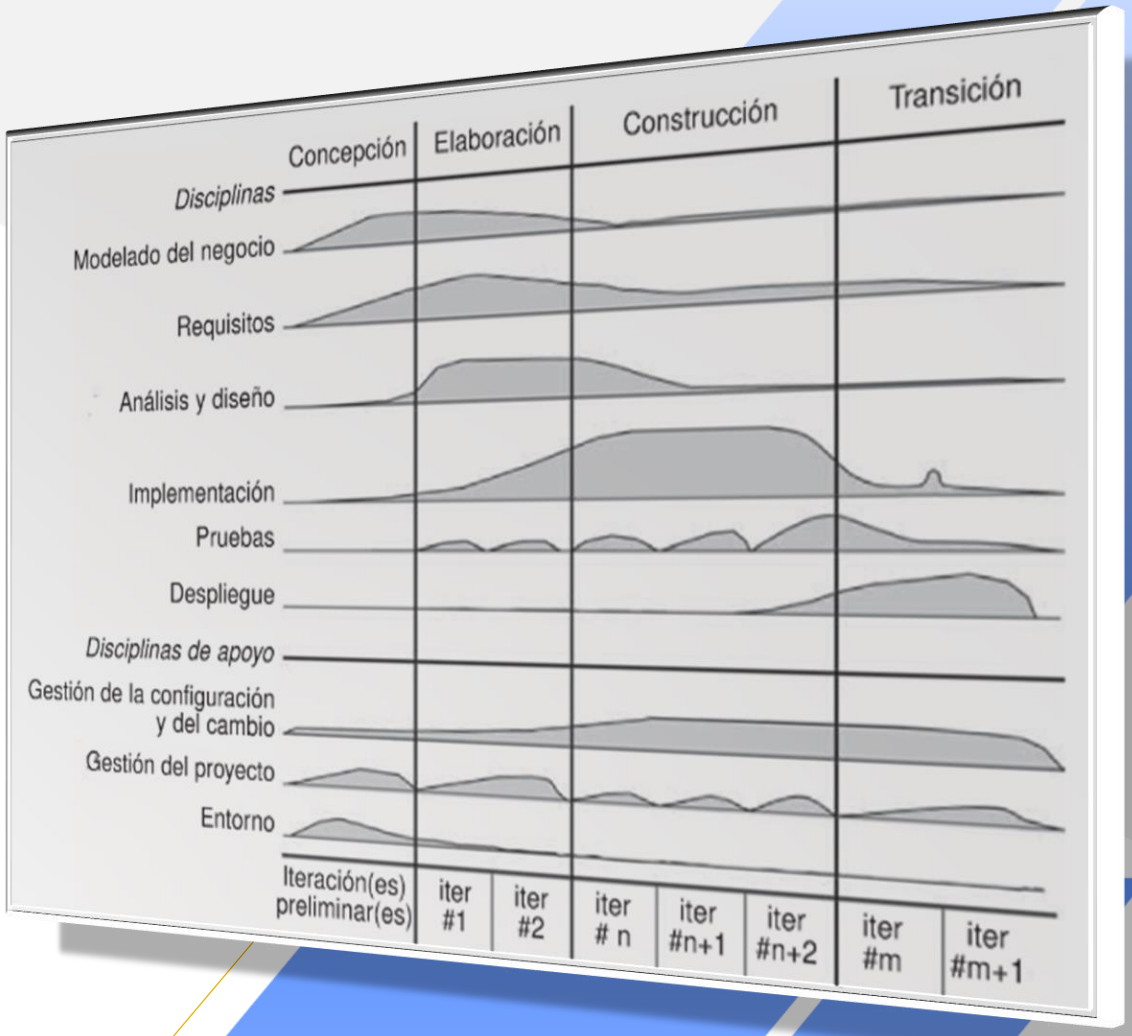
Vista de Casos de Uso: Describen el comportamiento del sistema desde la perspectiva de lo usuario.

Proceso Unificado de Rational

- UML es independiente del proceso (no está ligado a ningún ciclo de vida/desarrollo)
- Pero se puede utilizar con cualquiera que cumpla:
  - Dirigido por casos de uso (la vista del usuario)
  - Centrado en la arquitectura
  - Iterativo e incremental
- Por eso se lleva bien con el Proceso Unificado de Rational (RUP o PU):
  - Iterativo
  - Centrado en la arquitectura
  - Dirigido por CdU
  - Soporta técnicas orientadas a objetos
  - Promueve el control de calidad y la gestión de riesgo **dentro del proceso**

Proceso Unificado de Rational

- 4 Fases (intervalos entre hitos):
  - Concepción: Visión, alcance, plan inicial del proyecto
  - Elaboración: Diseñar, implementar y probar
  - Construcción: Versión operativa del sistema
  - Transición: Entregar el sistema a los usuarios finales
- 9 Disciplinas:
  - Modelado del negocio
  - Requisitos
  - Análisis y diseño
  - Implementación
  - Pruebas
  - Despliegue
  - Gestión de configuraciones
  - Gestión del proyecto
  - Entorno

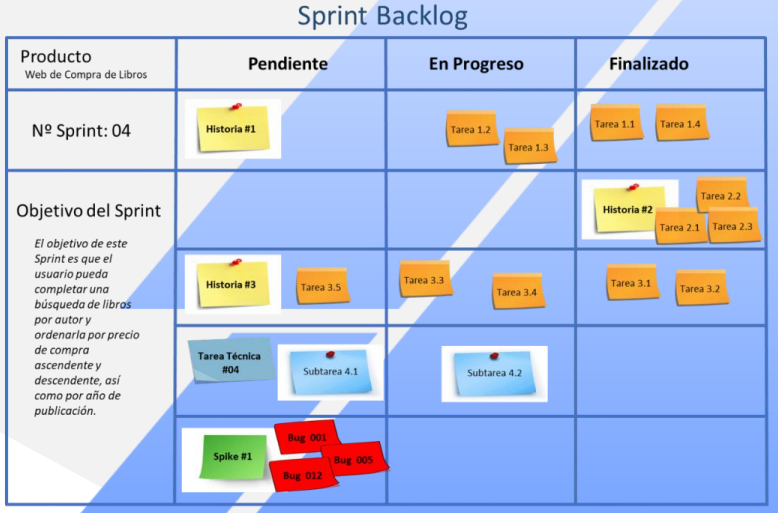


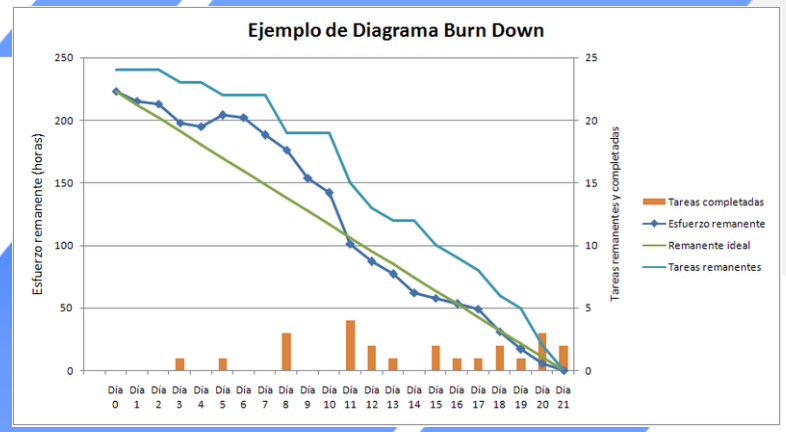
Metodologías ágiles: Scrum

- **Enfoque Ágil**: Scrum es un enfoque ágil para la gestión de proyectos que se centra en la flexibilidad, la colaboración y la entrega incremental de valor.
- **Roles Definidos**: Scrum Master (facilitador), Product Owner (definir los objetivos del proyecto) y el Equipo de Desarrollo (encargado de realizar el trabajo).
- **Iterativo e Incremental**: El trabajo se divide en "Sprints", que son períodos de tiempo fijos durante los cuales se desarrolla y entrega un conjunto de funcionalidades.
- **Product Backlog**: Se mantiene una lista de elementos llamada "Product Backlog" que contiene todas las características, mejoras y tareas pendientes.
- **Reuniones Regulares**: Reunión de Planificación del Sprint, Reunión Diaria, Reunión de Revisión del Sprint. Mantiene al equipo sincronizado y enfocado.

Metodologías ágiles: Scrum

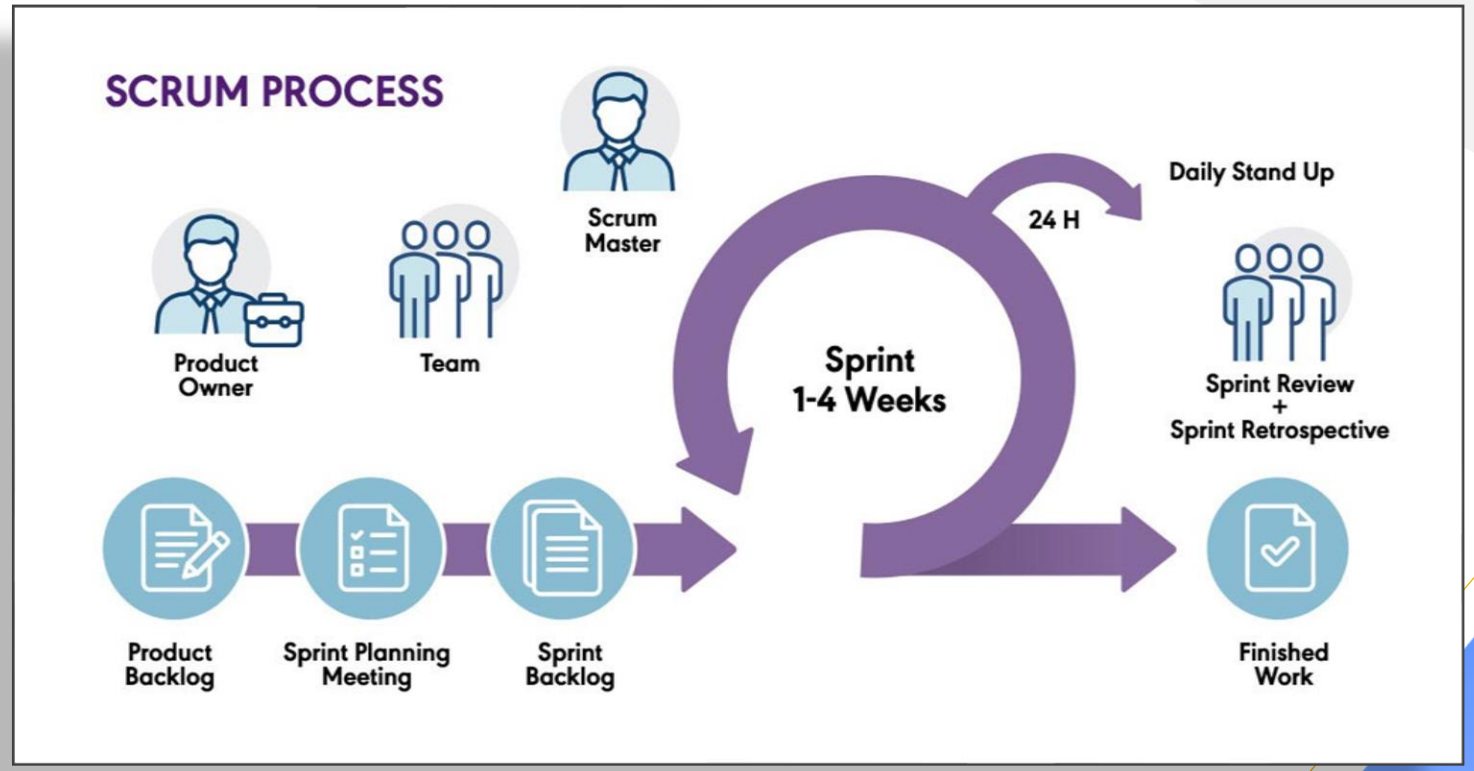
- **Transparencia**: Scrum promueve la transparencia en la comunicación y el seguimiento del progreso del proyecto a través de tableros Kanban y gráficos burndown.
- **Inspección y Adaptación**: Scrum enfatiza la inspección y adaptación continua. Al final de cada Sprint, se revisa el trabajo realizado y se ajusta el plan en consecuencia.
- **Entrega de Valor Temprano**: Scrum se enfoca en la entrega temprana y frecuente de características funcionales para obtener retroalimentación rápida del cliente.





Metodologías ágiles: Scrum

- **Colaboración y Autoorganización**: Fomenta la toma de decisiones conjuntas y la responsabilidad compartida.
- **Amplia Aplicabilidad**: Se ha aplicado con éxito en una amplia variedad de industrias y tipos de proyectos.



Video en youtube: Scrum en 10 minutos
https://www.youtube.com/watch?v=PlLHc60egiQ