



TECNOLOGÍA DE LA INFORMACIÓN EN LAS ORGANIZACIONES

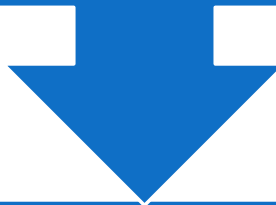
Cursada 2022



METODOLOGÍAS Y HERRAMIENTAS
PARA EL DESARROLLO DE SISTEMAS

METODOLOGÍAS Y HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS

Metodología: conjunto de métodos que se utilizan para cubrir las actividades de un proceso determinado.



Metodología de Desarrollo de Sistemas: conjunto de métodos que brindan soporte a cada una de las actividades dentro de las fases de un proyecto de sistemas.

Metodologías Estructuradas

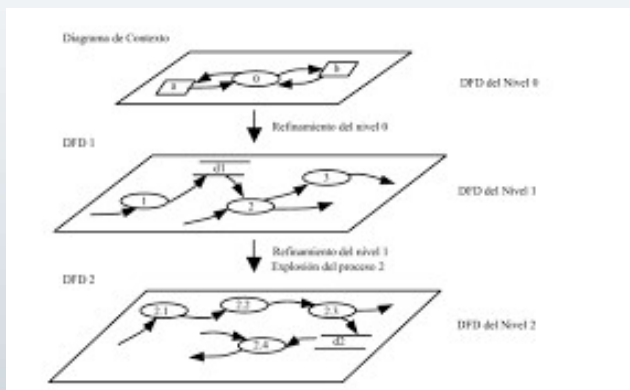
RUP (Rational Unified Process)

Desarrollo orientado a Objetos

Métodos Ágiles

Reingeniería del Software

METODOLOGÍAS ESTRUCTURADAS



Técnicas orientadas a los procesos más que a los datos.

Son descendentes: desde el nivel más alto y abstracto hasta el nivel más bajo y detallado.

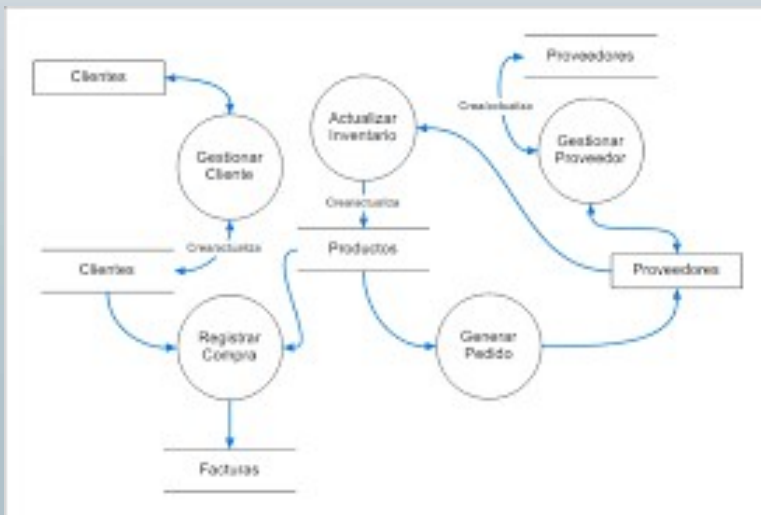
Utilizadas para el enfoque de ciclo de vida tradicional.

Incluyen:

Análisis estructurado → DFD, Especificaciones de Procesos.

Diseño estructurado → Diagrama de estructura

Programación estructurada → Diagrama de Flujo de Sistemas, Estructuras de control (secuencia, selección, iteración), Modularización

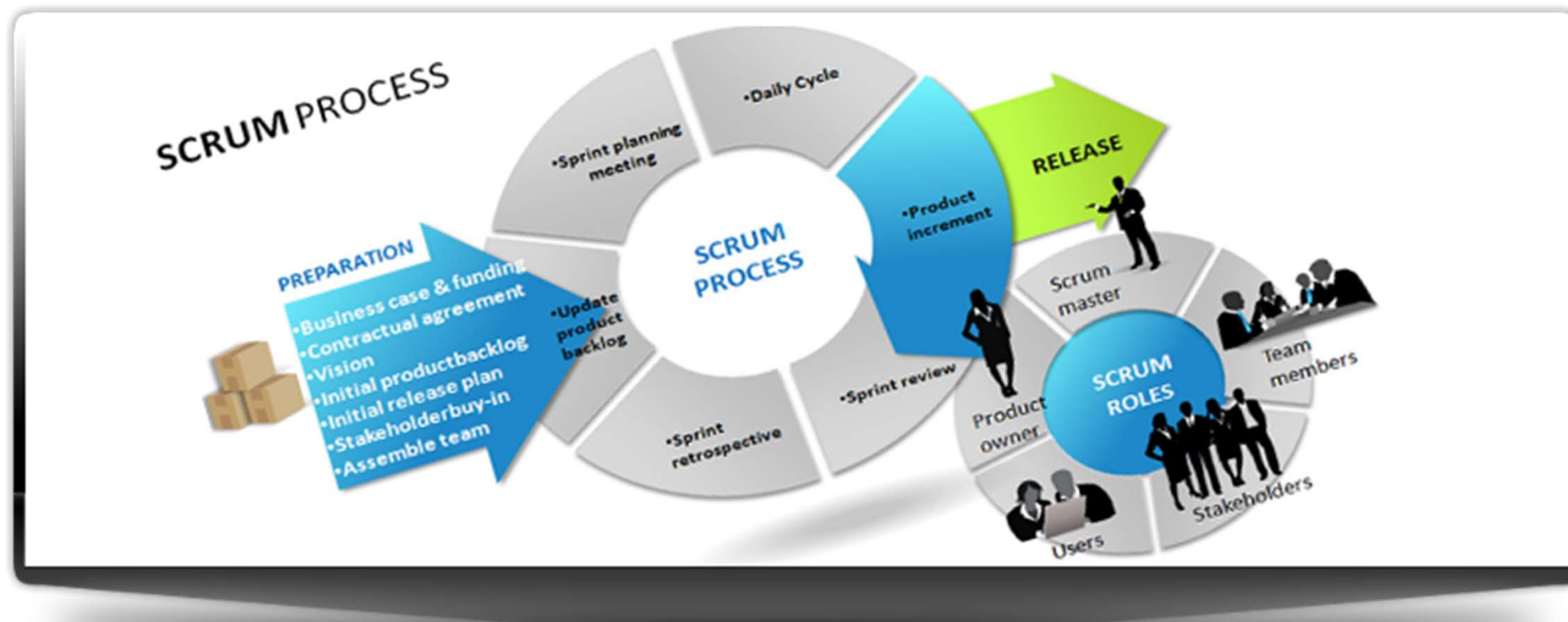


MÉTODOS AGILES



- ✓ Proceso de Creación de Sistemas funcionales en tiempo muy corto.
- ✓ Proceso no secuencial. Partes clave del desarrollo se realizan en paralelo.
- ✓ Utilizadas para el enfoque de prototipos y con herramientas de cuarta generación.
- ✓ Colaboración estrecha entre usuarios y especialistas de sistemas.
- ✓ No se genera casi documentación.
- ✓ Técnicas utilizables: diseño conjunto de aplicaciones (Joint Application Design –JAD-, Scrum, XP).

MÉTODOS AGILES



RUP(RATIONAL UNIFIED PROCESS)

✓ Metodología que divide el proceso en 4 fases:

1. Inicio,
2. Elaboración,
3. Construcción
4. Transición.

✓ Utilizable para cualquier tipo de proyecto. Cada fase tiene sus actividades asociadas

✓ Metodología iterativa con desarrollo incremental o en cascada

✓ La documentación se basa en ciertos diagramas y para esto, utiliza el UML. Ejemplos de diagramas:

- Para el análisis: Casos de Uso, Diagramas de estados –
- Para el diseño: Diagramas de clase, Diagramas de componentes, Diagrama de comunicaciones, etc.)



DESARROLLO ORIENTADO A OBJETOS



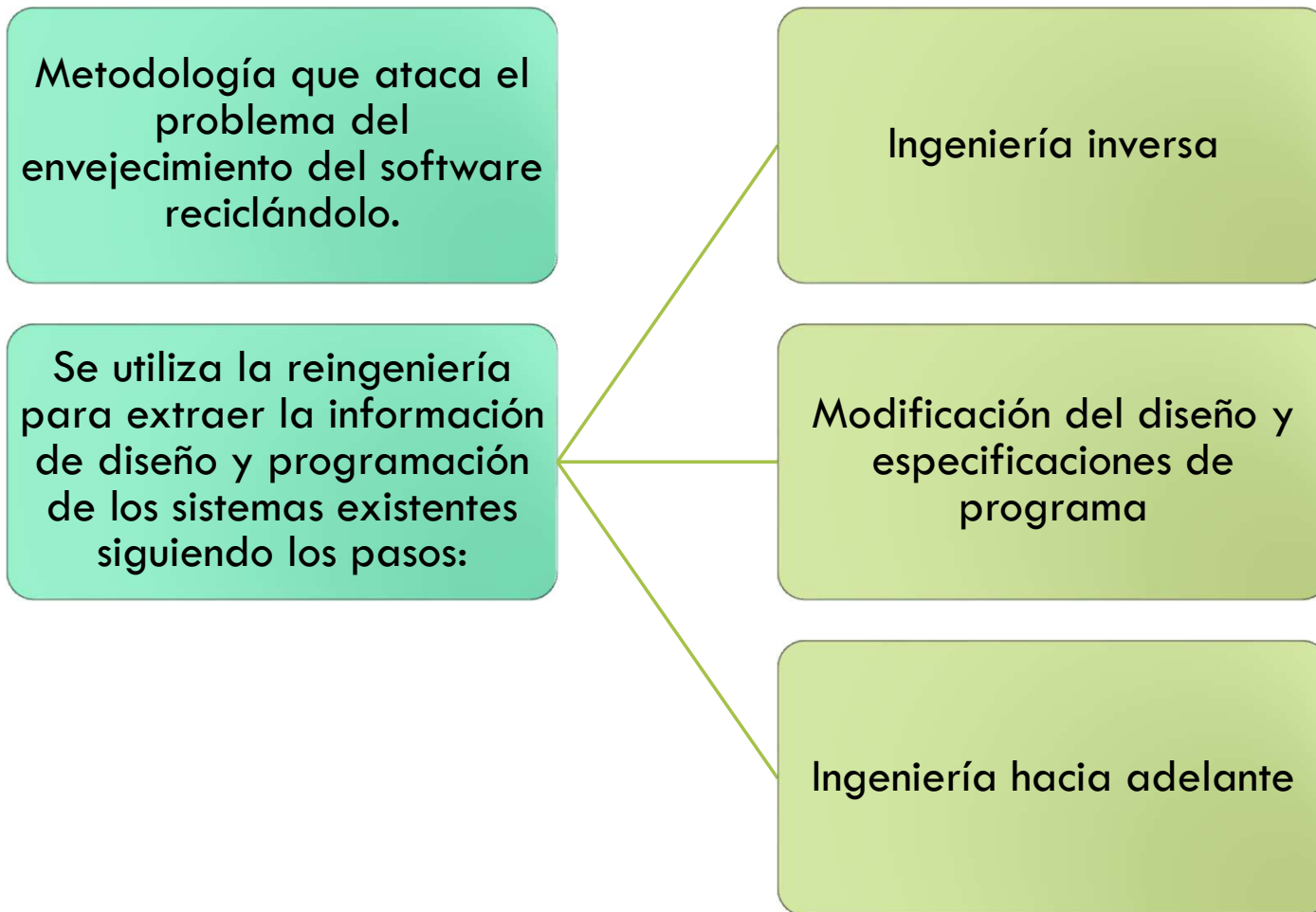
Objeto como unidad básica que encapsula sus datos y acciones que se pueden realizar sobre los mismos. Los objetos se relacionan entre si.

Los objetos tienen la característica que son reutilizables, se agrupan en clases y utilizan herencia.

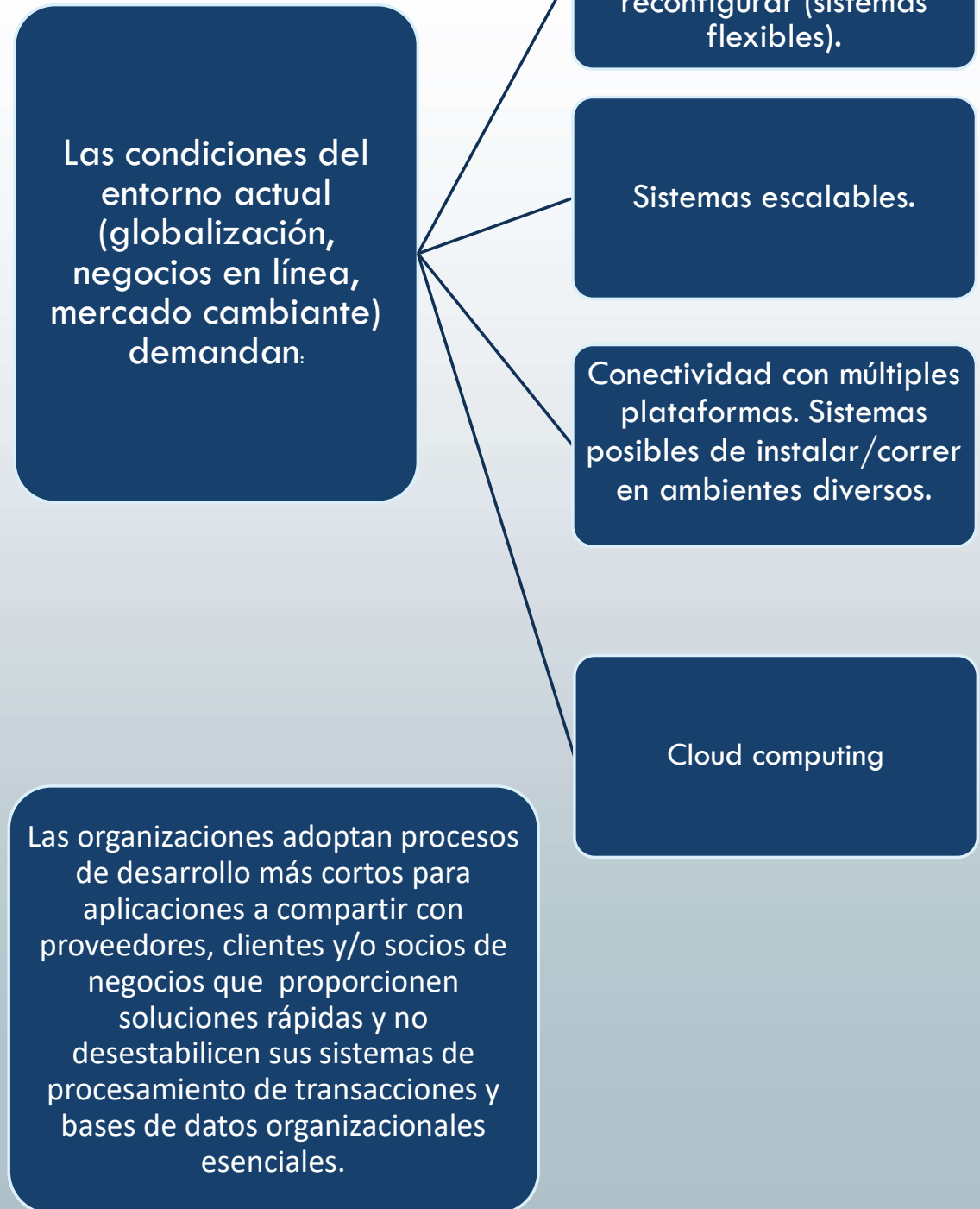
Utilizado para enfoque híbrido entre ciclo de vida tradicional y prototipos

Desarrollo de frameworks
→ Reusabilidad

REINGENIERÍA DE SOFTWARE



TENDENCIAS ACTUALES



CADENA DE
VALOR EN EL
DESARROLLO
DE SISTEMAS



ÉXITO Y FRACASO DE LOS SISTEMAS DE INFORMACIÓN

Los proyectos de Desarrollo de sistemas fracasan por:

- ✓ Mayores costos de lo previsto → Costos y Recursos
- ✓ No terminan en el tiempo estimado → Tiempos
- ✓ No son operables → Alcance.

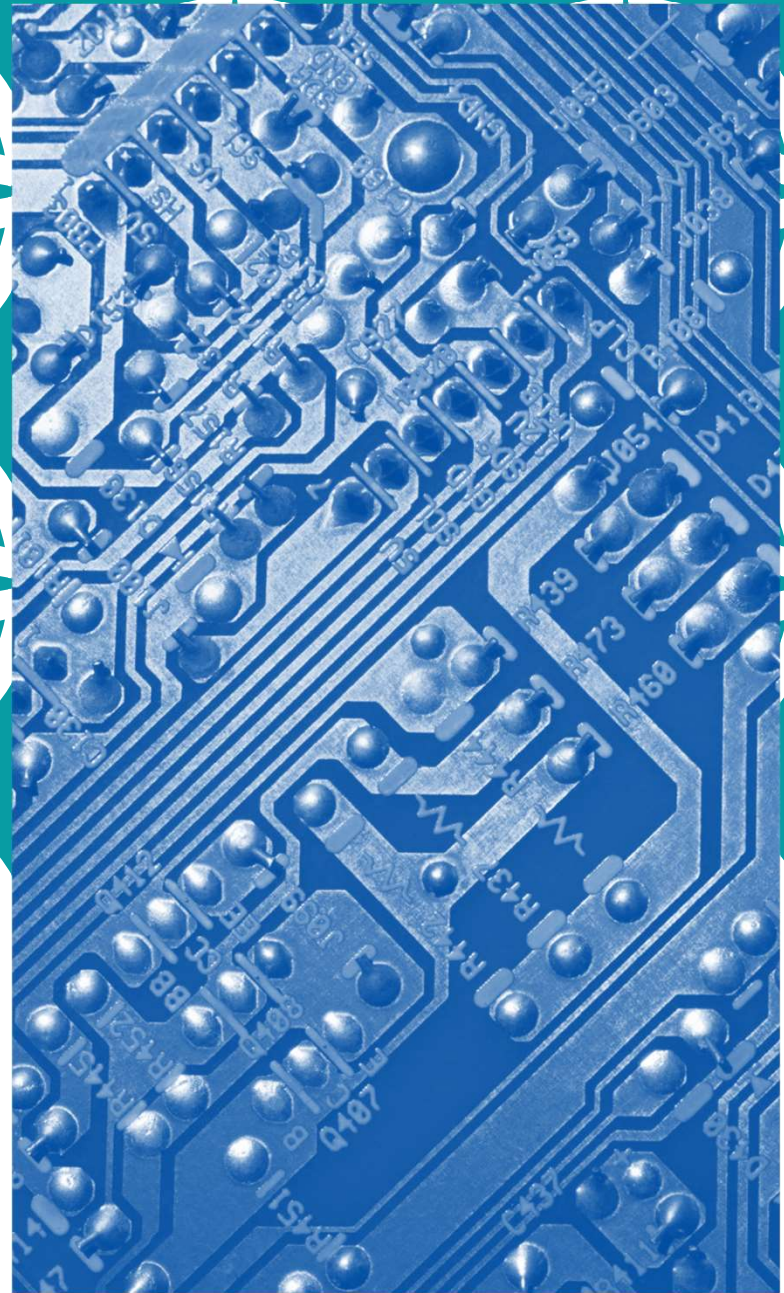
Principales áreas de problemas (origen en factores de organización):

- ✓ Diseño no compatible con la estructura, cultura y metas de la organización.
- ✓ Datos: Información ambigua, errónea, desglosada indebidamente o incompleta.
- ✓ Costos: Los gastos excesivos no pueden justificarse con el valor que el sistema proporciona al negocio.
- ✓ Operaciones: Información que no se proporciona en forma oportuna o eficiente por fallas en el procesamiento de la misma.

CULTURE
EATS STRATEGY
FOR BREAKFAST
AND TECHNOLOGY
FOR LUNCH
AND THEN...



GESTIÓN DE CONFIGURACIÓN DE SOFTWARE



GESTIÓN DE CONFIGURACIÓN DE SOFTWARE

Se refiere a las prácticas y herramientas transversales al desarrollo de software (a los requerimientos o el diseño en sí del programa), para atacar incumbencias como la:

Trazabilidad

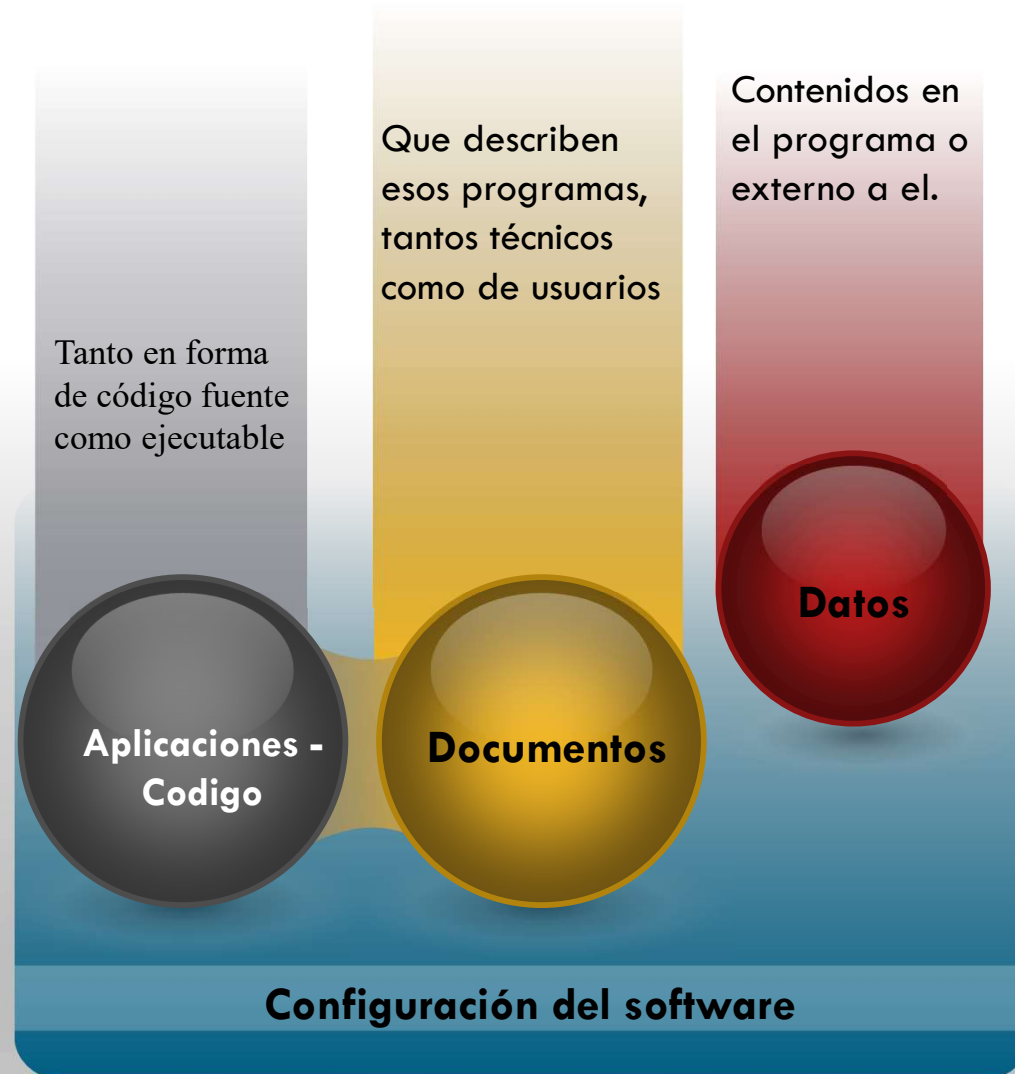
- ✓ *Gestión de Versiones*
- ✓ *Gestión de Cambios*
- ✓ *Gestión de Requerimientos*
- ✓ *Gestión de Incidencias.*

Reproducibilidad de releases

Comunicación

Interacción, coordinación e integración del trabajo de los diferentes miembros del equipo

Proceso de Gestión de Configuración de Software



Categorías del resultado del proceso de ing. del software

SISTEMA DE VERSIONADO DE CÓDIGO

Un Sistema de Versionado de Código (SVC) nos permite **compartir el código** fuente de nuestros desarrollos y a la vez **mantener un registro de los cambios** por los que va pasando.

En general va a ser la herramienta más importante y fundamental dentro del desarrollo. Veremos que también vamos a tener otras herramientas para publicar releases, documentación, para integración y control de calidad, pero todo va a depender de la herramienta SVC.

Si se pierden los releases, más allá de ser un inconveniente, se puede salvar en base al SVC.

SISTEMA DE VERSIONADO DE CÓDIGO

Se basan en versionar archivos y carpetas (repositorio)

Los desarrolladores utilizan un programa **cliente** que permite una serie de operaciones básicas:

- **obtener una copia** local de repositorio
- **publicar cambios** a un repositorio

Otras funciones:

- revisar los cambios
- obtener una copia de una **revisión** en particular
- deshacer cambios.
- marcar (taggear) una revisión
- ver quién, por qué y cuándo modifico cierto archivo/carpetas

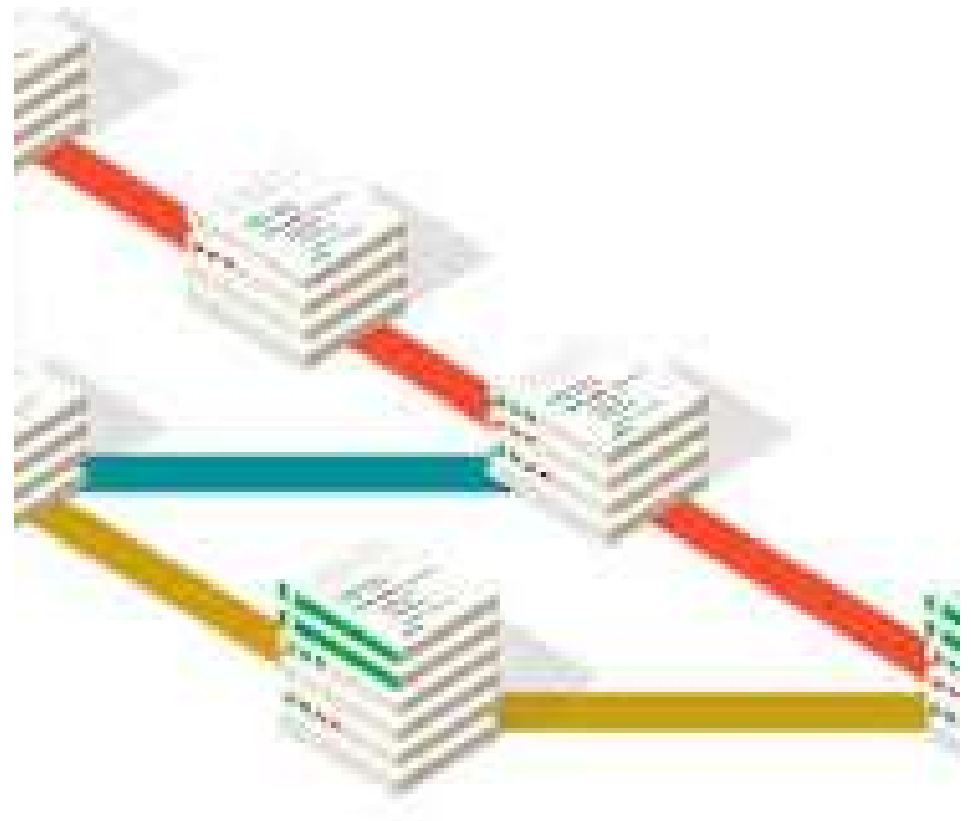
Estas operaciones dependen un poco del tipo de SCV que estemos utilizando.

TIPOS DE VERSIONADORES DE CÓDIGO

Sistemas

Centralizados: Son los más "tradicionales", por ejemplo SVN, CVS, etc.

Sistemas Distribuidos
(o descentralizados): son los que están en auge actualmente como: Git, Mercurial, Bazaar, etc.



PARA RECORDAR!!!

- ✓ *Software como producto/servicio*
- ✓ *Creación de sistemas*
 - *Tradicional*
 - *Prototipos*
 - *Desarrollo por usuarios finales*
 - *Compra de licencias*
 - *Terciarización*
- ✓ *Metodologías de desarrollo:*
 - *Metodologías Estructuradas*
 - *RUP (Rational Unified Process)*
 - *Desarrollo orientado a Objetos*
 - *Métodos Agiles*
 - *Reingeniería del Software*
- ✓ *Trazabilidad*
 - *Gestión de Versiones*
 - *Gestión de Cambios*
 - *Gestión de Requerimientos*
 - *Gestión de Incidencias*

