



## **UNIDAD 1**

- **Ciclo de vida del desarrollo de software.**
- **Modelos del desarrollo de software.**
- **Actividades involucradas.**



## Las etapas del proceso de desarrollo de software

- **Proceso de software:**

Un **proceso de software es una serie de actividades** relacionadas que conduce a la elaboración de un producto de software. Estas actividades pueden incluir el **desarrollo de software desde cero** en un lenguaje de programación.

*El nuevo software empresarial con frecuencia ahora se desarrolla extendiendo y modificando los sistemas existentes, o configurando e integrando el software comercial o componentes del sistema.*



## Las etapas del proceso de desarrollo de software

Existen muchos diferentes procesos de software, pero todos deben incluir cuatro actividades que son fundamentales para la ingeniería de software:

1. Especificación del software: Tienen que definirse tanto la funcionalidad del software como las restricciones de su operación.
2. Diseño e implementación del software: Debe desarrollarse el software para cumplir con las especificaciones.
3. Validación del software: Hay que validar el software para asegurarse de que cumple lo que el cliente quiere.
4. Evolución del software: El software tiene que evolucionar para satisfacer las necesidades cambiantes del cliente.



## **Clasificación de los procesos de software:**

- Dirigidos por un plan: son aquellos donde todas las actividades del proceso se planean por anticipado y el avance se mide contra dicho plan.
- Procesos ágiles: la planeación es incremental y es más fácil modificar el proceso para reflejar los requerimientos cambiantes del cliente.



## Modelos de proceso de software:

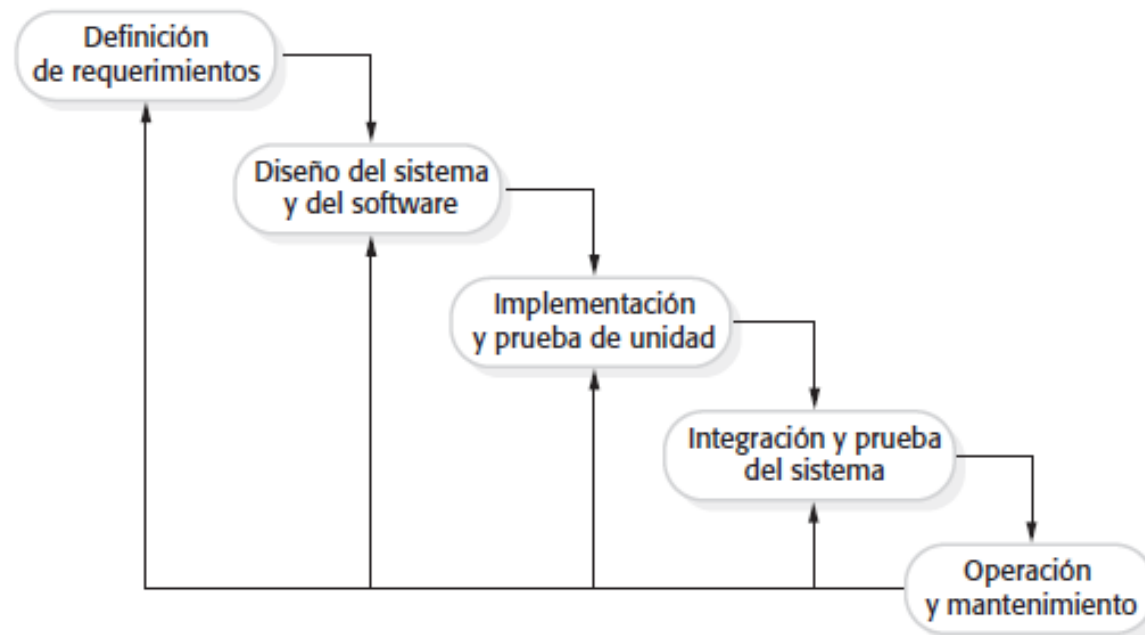
Un modelo de proceso de software es una representación simplificada de este proceso. Cada modelo del proceso representa a otro desde una particular perspectiva.

1. El modelo en cascada.
2. Desarrollo incremental.
3. Ingeniería de software orientada a la reutilización.



## Modelos de proceso de software:

1. El modelo en cascada: toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y, luego, los representa como fases separadas del proceso, tal como especificación de requerimientos, diseño de software, implementación, pruebas, etcétera.





## Modelos de proceso de software:

1. El modelo en cascada: Es un ejemplo de un proceso dirigido por un plan; se debe planear y programar todas las actividades del proceso, antes de comenzar a trabajar con ellas.
  - ✓ *Análisis y definición de requerimientos*
  - ✓ *Diseño del sistema y del software*
  - ✓ *Implementación y prueba de unidad*
  - ✓ *Integración y prueba de sistema*
  - ✓ *Operación y mantenimiento*



## Modelos de proceso de software:

En principio, el resultado de cada fase consiste en uno o más documentos que se autorizaron (“firmaron”). La siguiente fase no debe comenzar sino hasta que termine la fase previa. En la práctica, dichas etapas se traslapan y se nutren mutuamente de información.

Durante el diseño se identifican los problemas con los requerimientos. En la codificación se descubren problemas de diseño, y así sucesivamente. El proceso de software no es un simple modelo lineal, sino que implica retroalimentación de una fase a otra.

Entonces, es posible que los documentos generados en cada fase deban modificarse para reflejar los cambios que se realizan.





## Modelos de proceso de software:

Debido a los costos de producción y aprobación de documentos, las iteraciones suelen ser onerosas e implicar un rediseño significativo. Por lo tanto, después de un pequeño número de iteraciones, es normal detener partes del desarrollo, como la especificación, y continuar con etapas de desarrollo posteriores. Los problemas se dejan para una resolución posterior, se ignoran o se programan. Este freno prematuro de los requerimientos quizá signifique que el sistema no hará lo que el usuario desea. También podría conducir a sistemas mal estructurados conforme los problemas de diseño se evadan con la implementación de “trucos”.

Durante la fase final del ciclo de vida (operación y mantenimiento), el software se pone en servicio. Se descubren los errores y las omisiones en los requerimientos originales del software. Surgen los errores de programa y diseño, y se detecta la necesidad de nueva funcionalidad. Por lo tanto, el sistema debe evolucionar para mantenerse útil.

Hacer tales cambios puede implicar la repetición de etapas anteriores del proceso.



## Modelos de proceso de software:

El modelo en cascada es consecuente con otros modelos del proceso de ingeniería y en cada fase se produce documentación. Esto hace que el proceso sea visible, de modo que los administradores monitoricen el progreso contra el plan de desarrollo.

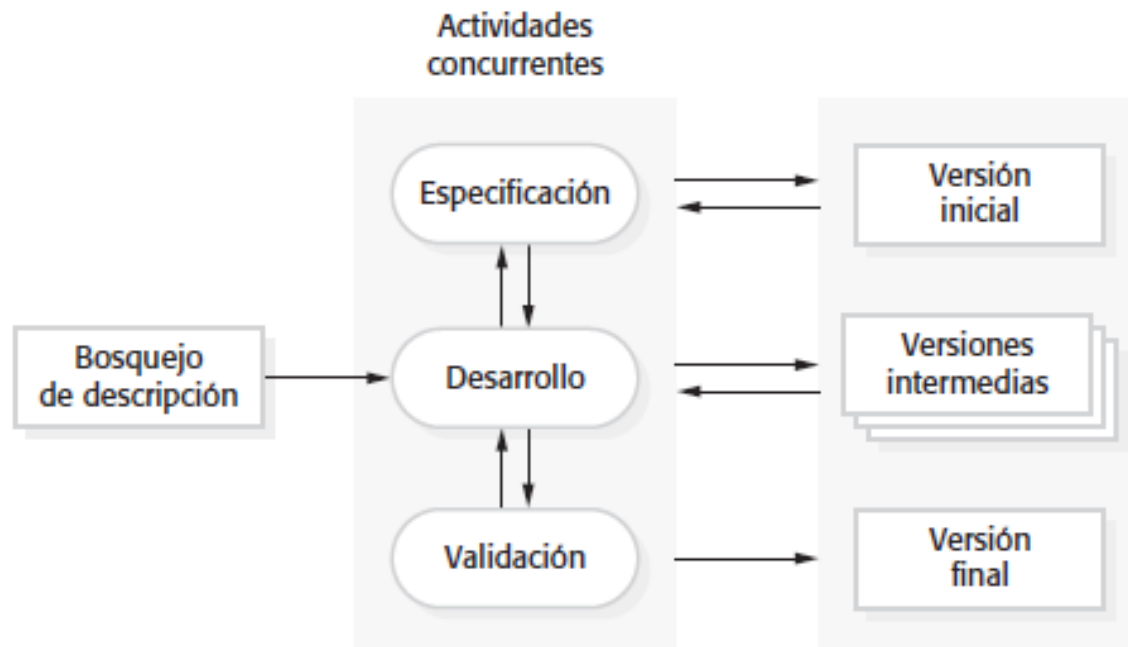
Su principal problema es la partición inflexible del proyecto en distintas etapas. Tienen que establecerse compromisos en una etapa temprana del proceso, lo que dificulta responder a los requerimientos cambiantes del cliente.

***El modelo en cascada sólo debe usarse cuando los requerimientos se entiendan bien y sea improbable el cambio radical durante el desarrollo del sistema.***



## Modelos de proceso de software:

2.- Desarrollo incremental: Este enfoque vincula las actividades de especificación, desarrollo y validación. El sistema se desarrolla como una serie de versiones (incrementos), y cada versión añade funcionalidad a la versión anterior.





## Modelos de proceso de software:

El desarrollo incremental se basa en la idea de diseñar una implementación inicial, exponer ésta al comentario del usuario, y luego desarrollarla en sus diversas versiones hasta producir un sistema adecuado.

Las actividades de especificación, desarrollo y validación están entrelazadas en vez de separadas, con rápida retroalimentación a través de las actividades.

El desarrollo de software incremental, que es una parte fundamental de los enfoques ágiles, es mejor que un enfoque en cascada para la mayoría de los sistemas empresariales, de comercio electrónico y personales.

Resulta más barato y fácil realizar cambios en el software conforme éste se diseña.



Comparado con el modelo en cascada, el desarrollo incremental tiene tres beneficios importantes:

1. Se reduce el costo de adaptar los requerimientos cambiantes del cliente.
2. Es más sencillo obtener retroalimentación del cliente sobre el trabajo de desarrollo que se realizó.
3. Es posible que sea más rápida la entrega e implementación de software útil al cliente.



## Modelos de proceso de software:

El desarrollo incremental ahora es en cierta forma el enfoque más común para el desarrollo de sistemas de aplicación. Este enfoque **puede estar basado en un plan, ser ágil o, más usualmente, una mezcla de dichos enfoques.**

En un enfoque basado en un plan se identifican por adelantado los incrementos del sistema; si se adopta un enfoque ágil, se detectan los primeros incrementos, aunque el desarrollo de incrementos posteriores depende del avance y las **prioridades** del cliente.

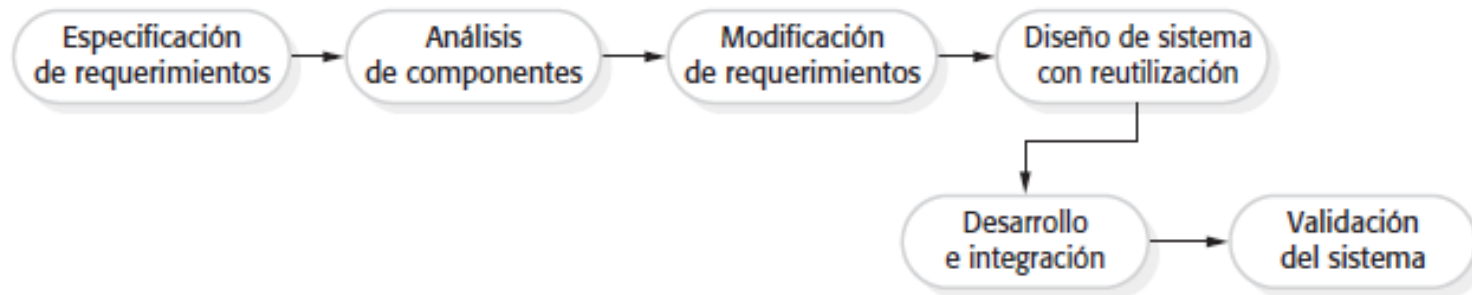
El enfoque incremental tiene dos problemas:

1. El proceso no es visible. Los administradores necesitan entregas regulares para medir el avance.
2. La estructura del sistema tiende a degradarse conforme se tienen nuevos incrementos.



## Modelos de proceso de software:

3.- Ingeniería de software orientada a la reutilización: Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en la integración de estos componentes en un sistema, en vez de desarrollarlo desde cero.





## Modelos de proceso de software:

En la mayoría de los proyectos de software hay cierta reutilización de software. Con frecuencia de manera informal, cuando las personas que trabajan en el proyecto conocen diseños o códigos que son similares a lo que se requiere. Los buscan, los modifican según se necesite y los incorporan en sus sistemas.

- La reutilización informal ocurre independientemente del proceso de desarrollo que se emplee.
- Este enfoque se utiliza ampliamente.
- Se apoya en una gran base de componentes de software.
- Cuentan con la integración de marcos para la composición de dichos componentes.





## Modelos de proceso de software:

Aunque la etapa inicial de especificación de requerimientos y la etapa de validación se comparan con otros procesos de software en un proceso orientado a la reutilización, las etapas intermedias son diferentes. Dichas etapas son:

1. *Análisis de componentes*
2. *Modificación de requerimientos*
3. *Diseño de sistema con reutilización*
4. *Desarrollo e integración*



## Modelos de proceso de software:

Existen tres tipos de componentes de software que pueden usarse en un proceso orientado a la reutilización:

1. Servicios Web que se desarrollan en concordancia para atender servicios estándares y que están disponibles para la invocación remota.
2. Colecciones de objetos que se desarrollan como un paquete para su integración con un marco de componentes como .NET o J2EE.
3. Sistemas de software independientes que se configuran para usar en un entorno particular.



## Modelos de proceso de software:

Las cuatro actividades básicas de proceso de especificación, desarrollo, validación y evolución se organizan de diversa manera en diferentes procesos de desarrollo.

En el modelo en **cascada** se organizan en **secuencia**, mientras que se **entrelazan** en el desarrollo **incremental**.

La forma en que se llevan a cabo estas actividades depende del tipo de software, del personal y de la inclusión de estructuras organizativas.



## Modelos de proceso de software:

### 1. Especificación del software:

La especificación del software o la ingeniería de requerimientos consisten en el proceso de comprender y definir qué servicios se requieren del sistema, así como la identificación de las restricciones sobre la operación y el desarrollo del sistema.

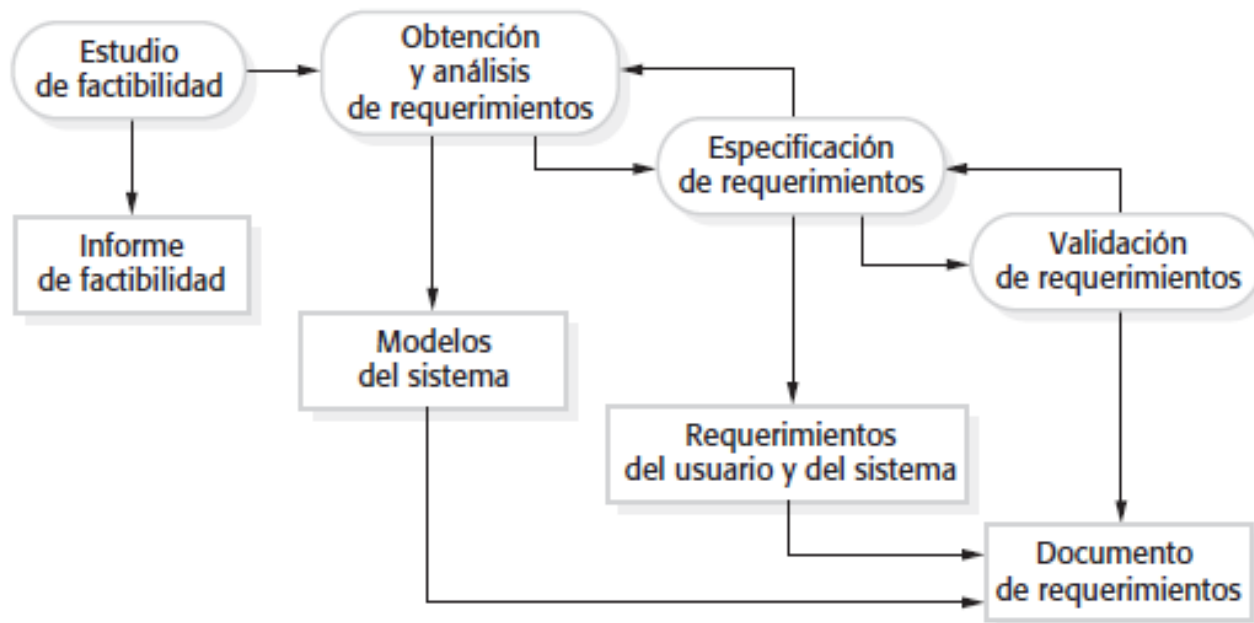
El proceso de ingeniería de requerimientos se enfoca en producir un documento de requerimientos convenido que especifique los requerimientos de los interesados que cumplirá el sistema.

Por lo general, los requerimientos se presentan en dos niveles de detalle. Los usuarios finales y clientes necesitan un informe de requerimientos de alto nivel; los desarrolladores de sistemas precisan una descripción más detallada del sistema.



## Modelos de proceso de software:

### 1. Especificación del software:





## Modelos de proceso de software:

Existen cuatro actividades principales en el proceso de ingeniería de requerimientos:

1. *Estudio de factibilidad*
2. *Obtención y análisis de requerimientos*
3. *Especificación de requerimientos*
4. *Validación de requerimientos*



## Modelos de proceso de software:

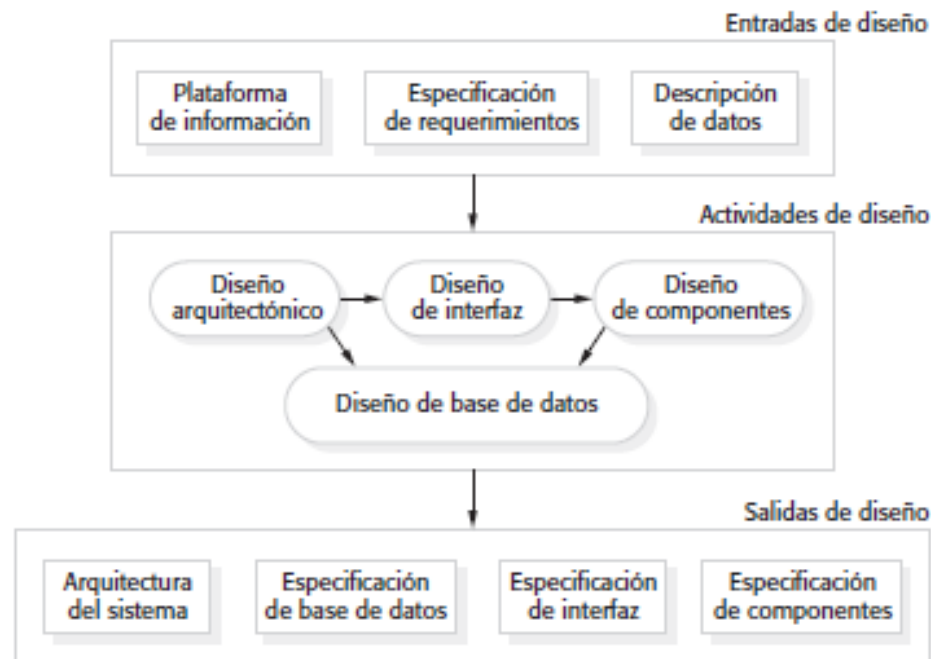
### 2. Diseño e implementación del software

La etapa de diseño y desarrollo del software corresponde al proceso de convertir una especificación del sistema en un sistema ejecutable. Siempre incluye procesos de diseño y programación de software, aunque también puede involucrar la corrección en la especificación del software, si se utiliza un enfoque incremental de desarrollo.



## Modelos de proceso de software:

2.- Diseño e implementación del software: El diagrama sugiere que las etapas del proceso de diseño son secuenciales. De hecho, las actividades de proceso de diseño están vinculadas. En todos los procesos de diseño es inevitable la retroalimentación de una etapa a otra y la consecuente reelaboración del diseño.







## Modelos de proceso de software:

Las cuatro actividades que podrían formar parte del proceso de diseño para sistemas de información:

1. *Diseño arquitectónico*, aquí se identifica la estructura global del sistema, los principales componentes, sus relaciones y cómo se distribuyen.
2. *Diseño de interfaz*, en éste se definen las interfaces entre los componentes de sistemas.

Con una interfaz precisa, es factible usar un componente sin que otros tengan que saber cómo se implementó. Una vez que se acuerdan las especificaciones de interfaz, los componentes se diseñan y se desarrollan de manera concurrente.



## Modelos de proceso de software:

3. *Diseño de componentes*, en él se toma cada componente del sistema y se diseña cómo funcionará. Esto puede ser un simple dato de la funcionalidad que se espera implementar, y al programador se le deja el diseño específico. Como alternativa, habría una lista de cambios a realizar sobre un componente que se reutiliza o sobre un modelo de diseño detallado.
4. *Diseño de base de datos*, donde se diseñan las estructuras del sistema de datos y cómo se representarán en una base de datos. De nuevo, el trabajo aquí depende de si una base de datos se reutilizará o se creará una nueva.



## Modelos de proceso de software:

3.- Validación de software: La verificación y validación, se crea para mostrar que un sistema cumple tanto con sus especificaciones como con las expectativas del cliente.

Las pruebas del programa, donde el sistema se ejecuta a través de datos de prueba simulados, son la principal técnica de validación.

Esta última también puede incluir procesos de comprobación, como inspecciones y revisiones en cada etapa del proceso de software, desde la definición de requerimientos del usuario hasta el desarrollo del programa.



## Modelos de proceso de software:

Las etapas en el proceso de pruebas son:

1. *Prueba de desarrollo:* Las personas que desarrollan el sistema ponen a prueba los componentes que constituyen el sistema. Cada componente se prueba de manera independiente, es decir, sin otros componentes del sistema. Éstos pueden ser simples entidades, como funciones o clases de objeto, o agrupamientos coherentes de dichas entidades.
2. *Pruebas del sistema:* Los componentes del sistema se integran para crear un sistema completo. Este proceso tiene la finalidad de descubrir errores que resulten de interacciones no anticipadas entre componentes y problemas de interfaz de componente, así como de mostrar que el sistema cubre sus requerimientos funcionales y no funcionales.
3. *Pruebas de aceptación:* Ésta es la etapa final en el proceso de pruebas, antes de que el sistema se acepte para uso operacional. El sistema se pone a prueba con datos suministrados por el cliente del sistema, en vez de datos de prueba simulados.



## Modelos de proceso de software:

### 4. Evolución del software:

La flexibilidad de los sistemas de software es una de las razones principales por las que cada vez más software se incorpora en los sistemas grandes y complejos. Una vez tomada la decisión de fabricar hardware, resulta muy costoso hacer cambios a su diseño. Sin embargo, en cualquier momento durante o después del desarrollo del sistema, pueden hacerse cambios al software. Incluso los cambios mayores son todavía más baratos que los correspondientes cambios al hardware del sistema.

En la historia, siempre ha habido división entre el proceso de desarrollo del software y el proceso de evolución del software (mantenimiento de software).

Esta distinción entre desarrollo y mantenimiento es cada vez más irrelevante. Es muy difícil que cualquier sistema de software sea un sistema completamente nuevo, y tiene mucho más sentido ver el desarrollo y el mantenimiento como un continuo. En lugar de dos procesos separados, es más realista pensar en la ingeniería de software como un proceso evolutivo



## Modelos de proceso de software:

### 4. Evolución del software:

