

# MODELO EN CASCADA

Ocurre en cierto número limitado de nuevos esfuerzos de desarrollo, sólo cuando los requerimientos están bien definidos y tienen una estabilidad razonable.

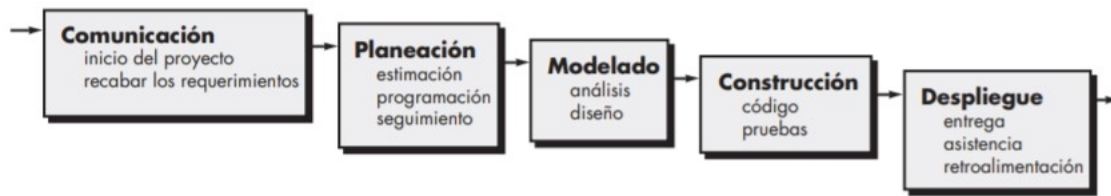
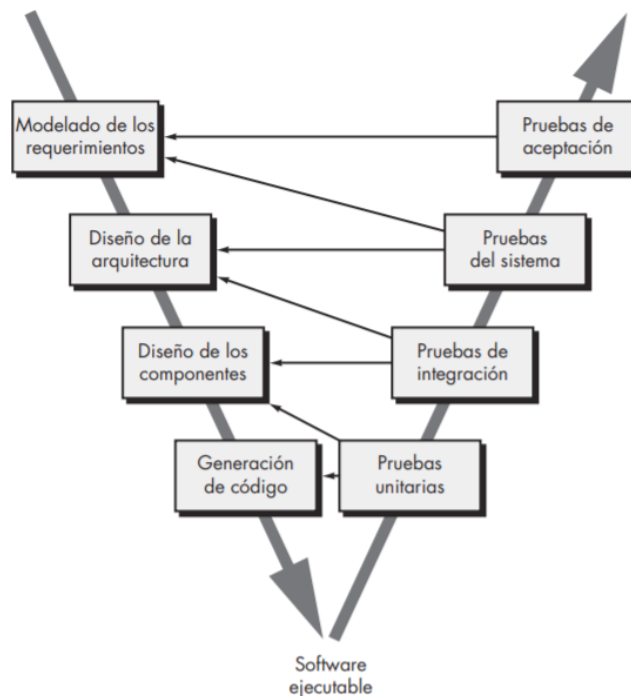


Figura 1: Modelo en cascada

Modelo de la cascada, o también llamado ciclo de vida clásico

Sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue.



A medida que el equipo de software avanza hacia abajo desde el lado izquierdo de la V, los requerimientos básicos del problema mejoran hacia representaciones técnicas cada vez más detalladas del problema y de su solución. Una vez que se ha generado el código, el equipo sube por el lado derecho de la V, y en esencia ejecuta una serie de pruebas (acciones para asegurar la calidad) que validan cada uno de los modelos creados cuando el equipo fue hacia abajo por el lado izquierdo.

El modelo de la cascada es el paradigma más antiguo de la ingeniería de software.

Problemas que en ocasiones surgen al aplicar el modelo de la cascada:

- Los proyectos reales no siempre siguen el flujo secuencial propuesto por el modelo.
- Es difícil para el cliente enunciar en forma explícita todos los requerimientos.
- El cliente debe tener paciencia.

## Modelos evolutivos

Los modelos evolutivos son iterativos, Se caracterizan por la manera en la que permiten desarrollar versiones cada vez más completas del software, los dos modelos comunes de proceso evolutivo son los siguientes:

### Hacer prototipos.

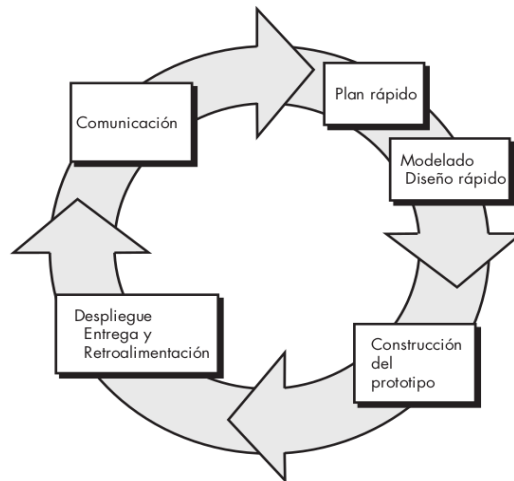
El paradigma de hacer prototipos ofrece un mejor enfoque, ayudara a los equipos a mejorar la comprensión de que se va a elaborar cuando los requerimientos no estén claros.

El paradigma de hacer prototipos tiene un proceso, el cual comienza con la comunicación, donde se reúne los participantes para definir los objetivo generales de software, se identifican los requerimientos que se conozca y detecta las áreas en las que es imprescindible una mayor definición, después de esto se plantea rápidamente una iteración para hacer el prototipo, y se lleva a cabo el modelado(diseño rápido), este se centra en la representación de los aspectos del software que serán visibles para los usuarios finales. El diseño rápido lleva a la construcción de un prototipo donde se entrega y es evaluado por los participantes, que dan retroalimentación para mejorar los requerimientos. Lo ideal es que el prototipo sirva como mecanismo para identificar los requerimientos del software, el prototipo sirve como “el primer sistema”, algunos prototipos se construyen para ser “desechables”, otros son evolutivos ya que poco a poco se transforman en el sistema real , a los ingenieros de software les gusta el paradigma de hacer prototipos por que ellos siente que así los usuarios adquieren la sensación del sistema real, y los desarrolladores logran construir algo de inmediato.

Algunas razones por las cual el hacer prototipos llega a ser algo problemático son las siguientes:

1. Los participantes ven lo que parece ser una versión funcional del software, sin darse cuenta de que el prototipo se obtuvo de manera caprichosa; no perciben que en la prisa por hacer que funcionara, usted no consideró la calidad general del software o la facilidad de darle mantenimiento a largo plazo.
2. Es frecuente que llegue a compromisos respecto de la implementación a fin de hacer que el prototipo funcione rápido. Quizá utilice un sistema operativo inapropiado.

Hacer prototipos es un paradigma eficaz para la ingeniería de software. La clave es definir desde el principio las reglas del juego, es decir, todos los participantes deben estar de acuerdo en que el prototipo sirva como el mecanismo para definir los requerimientos esto hará que la ingeniería del software real con la mirada puesta en la calidad.



## El modelo espiral.

El modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Un modelo en espiral es dividido por el equipo de software en un conjunto de actividades estructurales.

En el proceso evolutivo el equipo de software realiza actividades implícitas en un circuito alrededor de la espiral en el sentido horario, partiendo del centro.

El primer circuito alrededor de la espiral da como resultado el desarrollo de una especificación del producto; las vueltas sucesivas se usan para desarrollar un prototipo y, luego, versiones cada vez más sofisticadas del software, cada paso por la región de planeación da como resultado ajustes en el plan del proyecto, el costo y la programación de actividades se ajustan con base en la retroalimentación obtenida del cliente después de la entrega, con esto el gerente del proyecto ajusta el número planeado de iteraciones que se requieren para terminar el software.

El modelo espiral es un enfoque realista para el desarrollo de sistemas y de software a gran escala, el software evoluciona a medida que el proceso avanza, el desarrollador y cliente comprenden y reaccionan mejor ante los riesgos en cada nivel de evolución. El modelo espiral usa los prototipos como mecanismo de reducción de riesgos, permite aplicar el enfoque de hacer prototipos en cualquier etapa de la evolución del producto.

