

Universidad Autónoma del Estado de México

Facultad de Ingeniería

•• Ingeniería en Computación

Teoría de Sistemas

Unidad III

Modelos de Procesos de Desarrollo

Elaboró:

M. en A. Silvia Edith Albarrán Trujillo

Octubre 2016

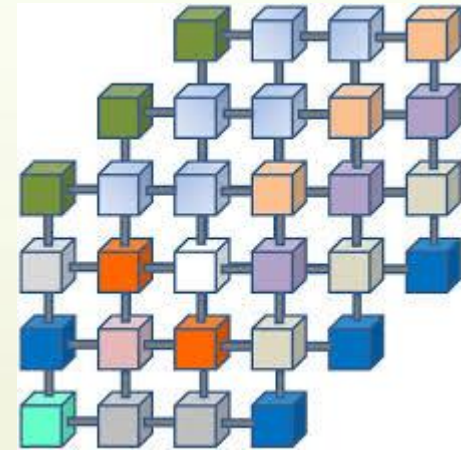
# Propósito de la Unidad de Aprendizaje

Que el estudiante obtenga una visión general de la ingeniería de software, así como de los principales modelos y ciclos de vida del software y de las metodologías asociadas a ellos, que sirva de soporte a los cursos de análisis y de diseño, mediante el estudio y seguimiento de una metodología en sus fases iniciales.



# Estructura de la Unidad de Aprendizaje

1. Introducción a la Ingeniería de Software
2. Tipos de organizaciones y de sistemas
- 3. Modelos de procesos de desarrollo**
4. Desarrollo ágil
5. El proceso unificado de desarrollo
6. Modelado organizacional o de negocios y Requisitos



# Guión para el Uso de este Material

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

- La información de esta presentación contiene ideas generales que serán explicadas en la clase.
- Para ampliar la información que se presenta en esta presentación se incluye al final un apartado de bibliografía.
- La presente contiene sólo información de la **Unidad III**,  
Titulada: **Modelos de procesos de desarrollo**
- Una vez concluida esta unidad el alumno tendrá familiaridad con los modelos de procesos de desarrollo: Modelo incremental (Incremental y DRA) y Modelos evolutivos (prototipo, espiral y concurrente).



# Contenido del material

1. Portada
2. Propósito de la Unidad de Aprendizaje
3. Estructura de la Unidad de Aprendizaje
4. Guión para uso de este material
5. Contenido del material
6. Objetivo de la Unidad III
7. Concepto de Ingeniería de Software
8. Proceso de software
9. Antecedentes
  - 9.1. Modelo Lineal Secuencial
10. Modelo Incremental
  - 10.1. Modelo incremental
  - 10.2. DRA
11. Modelos Evolutivos
  - 11.1. Prototipos
  - 11.2. Espiral
  - 11.3. Concurrente
12. Conclusiones
13. Referencias



# OBJETIVO DE LA UNIDAD III



El alumno conocerá los modelos de procesos de desarrollo: Modelo incremental (Incremental y DRA) y Modelos evolutivos (prototipo, espiral y concurrente), características, ventajas y desventajas

# Ingeniería de Software



Aplicación de un enfoque sistemático, disciplinado y cuantificable del desarrollo, operación y mantenimiento de software (Pressman, 2010).

# Proceso de Software

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

Conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del proyecto de software (Pressman, 2010).





# Antecedentes

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Ciclo de Vida Clásico Modelo Lineal Secuencial Modelo Cascada



# Modelo Lineal Secuencial

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Datos Históricos



- Desarrollo atribuido a Royce en 1970
- También conocido como “Ciclo de Vida Clásico” o “Modelo de Cascada”
- Paradigma más antiguo
- Modelo más utilizado durante la hegemonía del método estructurado

# Modelo Lineal Secuencial

## Diagrama de Fases

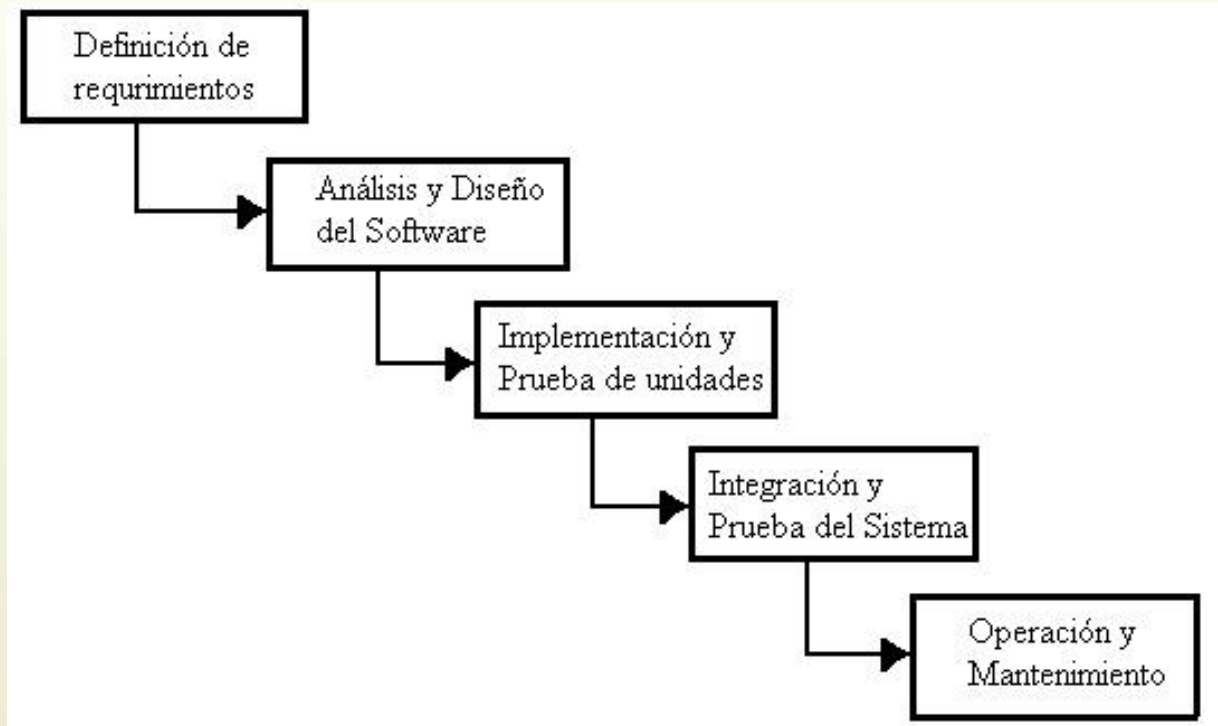


Imagen obtenida de: <http://procesosdesoftware.blogspot.mx/>. 4 de octubre de 2016

# Modelo Lineal Secuencial

## Descripción de las Fases

- 1. Definición de Requerimientos:** Requisitos que debe cumplir el software . Con la presencia del cliente.
- 2. Análisis y Diseño de Software:** Estructura de datos, la arquitectura del software, las representaciones de la interfaz y el detalle procedimental (algoritmo). Esbozo de lo solicitado y se documenta haciéndose parte del software
- 3. Implementación y Prueba de Unidades:** Se traduce el diseño para que sea comprensible por la máquina. Lo detallado del diseño



# Modelo Lineal Secuencial

## Descripción de las Fases

- 4. Integración y Prueba del Sistema:** centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en la detección de errores.
- 5. Operación y Mantenimiento:** acoplarse a los cambios en su entorno. Esto quiere decir que no se rehace el programa, sino que sobre la base de uno ya existente se realizan algunos cambios.



# Modelo Lineal Secuencial

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Ventajas

- Existe documentación en cada fase
- Estabilidad de operación
- Facilita la Gestión del desarrollo
- Debe utilizarse cuando se necesitan interconectar al sistema de información otros elementos de la empresa como hardware, persona y bases de datos.

# Modelo Lineal Secuencial

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Desventajas



- Una fase no inicia hasta que no termina la anterior
- Deben establecerse todos los requisitos al inicio del proceso
- El usuario deben esperar mucho para ver resultados
- Los errores de análisis y diseño son costosos de eliminar y se propagan a las fases siguientes
- Se utilizan muchos recursos en mantenimiento
- Los proyectos rara vez siguen un modelo secuencial

# Modelo Incremental

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

- Modelo Incremental
- DRA



# Modelo Incremental

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Datos Históricos



- Propuesto por Marlon Mills en 1980.
- Su principal objetivo fue reducir la repetición del proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema.
- Combina elementos del modelo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos.
- Entrega software “por partes funcionales más pequeñas”, pero reutilizables, llamadas incrementos.
- En general cada incremento se construye sobre aquel que ya fue entregado .

# Modelo Incremental

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Diagrama de Fases

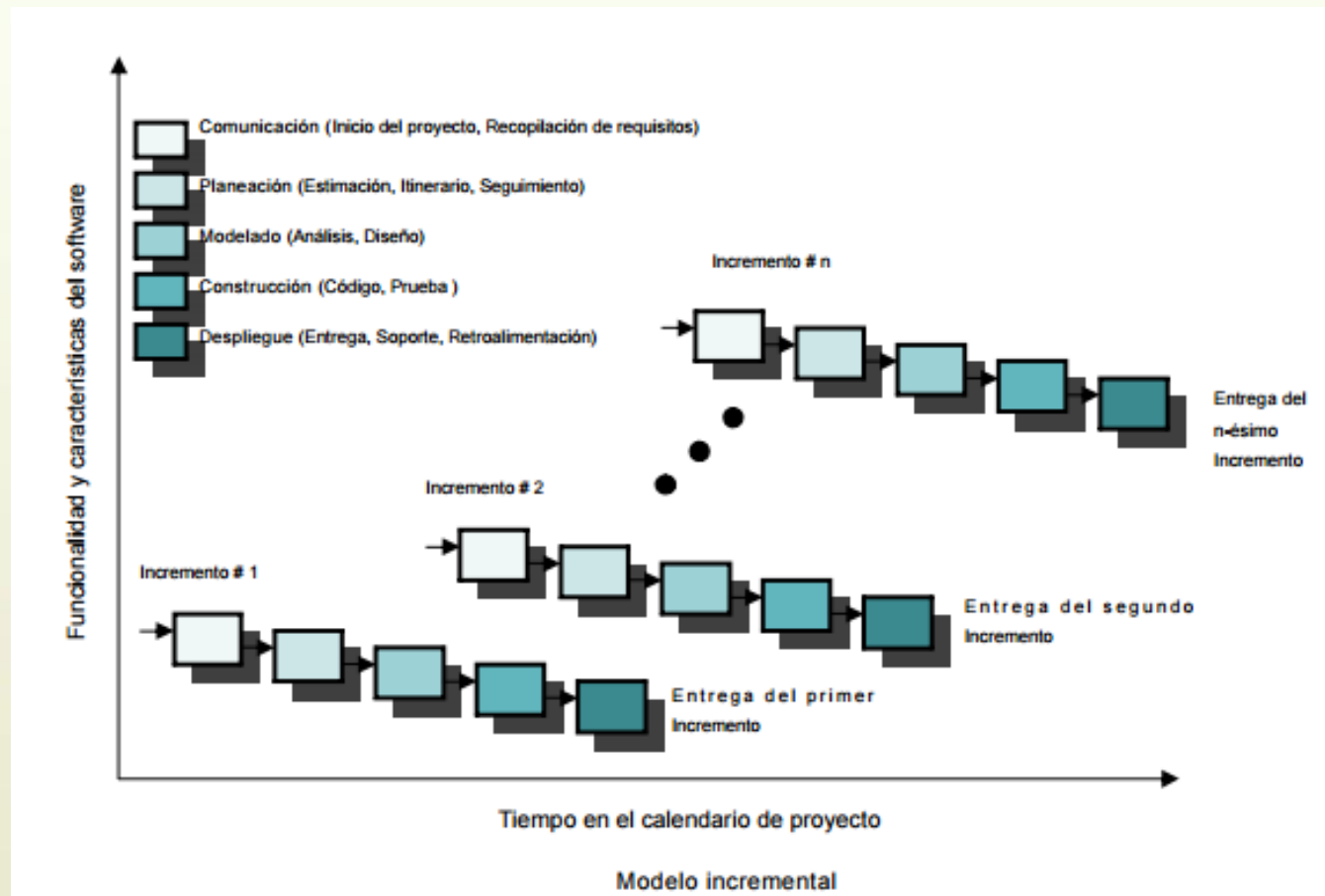


Imagen obtenida de Méndez, 2006.

# Modelo Incremental

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Descripción de las Fases

1. **Comunicación:** Inicio del proyecto, se realiza la recopilación de requerimientos con el cliente.
2. **Planeación:** Estimación de recurso y tiempos, se interpretan y validan los requisitos obtenidos.
3. **Modelado:** Se determina la estructura requerida para el incremento.

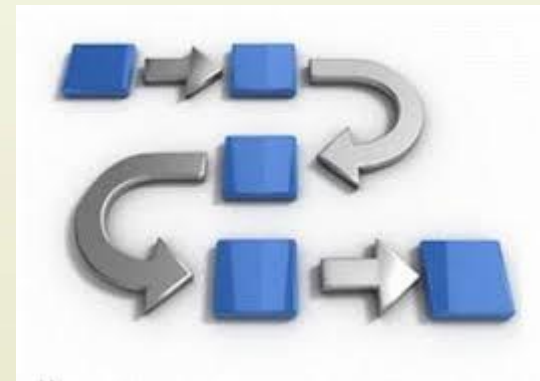


# Modelo Incremental

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Descripción de las Fases

4. **Construcción:** Elaboración del código y realización de las pruebas.
5. **Despliegue:** Entrega del producto y recomendaciones para su funcionamiento



# Modelo Incremental

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Ventajas



- Proporciona en forma progresiva más funcionalidad
- Útil cuando hay poco personal disponible
- Se reduce el tiempo de desarrollo inicial
- Es más sencillo implementar los cambios al acotar los incrementos
- El usuario puede ver los avances de manera paulatina
- Todas las ventajas del modelo lineal secuencial

# Modelo Incremental

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Desventajas

- No es recomendable para desarrollo de aplicaciones en tiempo real, de alto nivel de seguridad, con alto índice de riesgos
- Los primeros incrementos son versiones incompletas del producto final
- Requiere mucha planeación
- Requiere tiempo para la finalización del producto final
- Requiere el tiempo del cliente para cada incremento
- Falta de robustez
- Difícil de evaluar los incrementos
- Difícil de evaluar el avance.



# DRA (Desarrollo Rápido de Aplicaciones)

## Datos Históricos



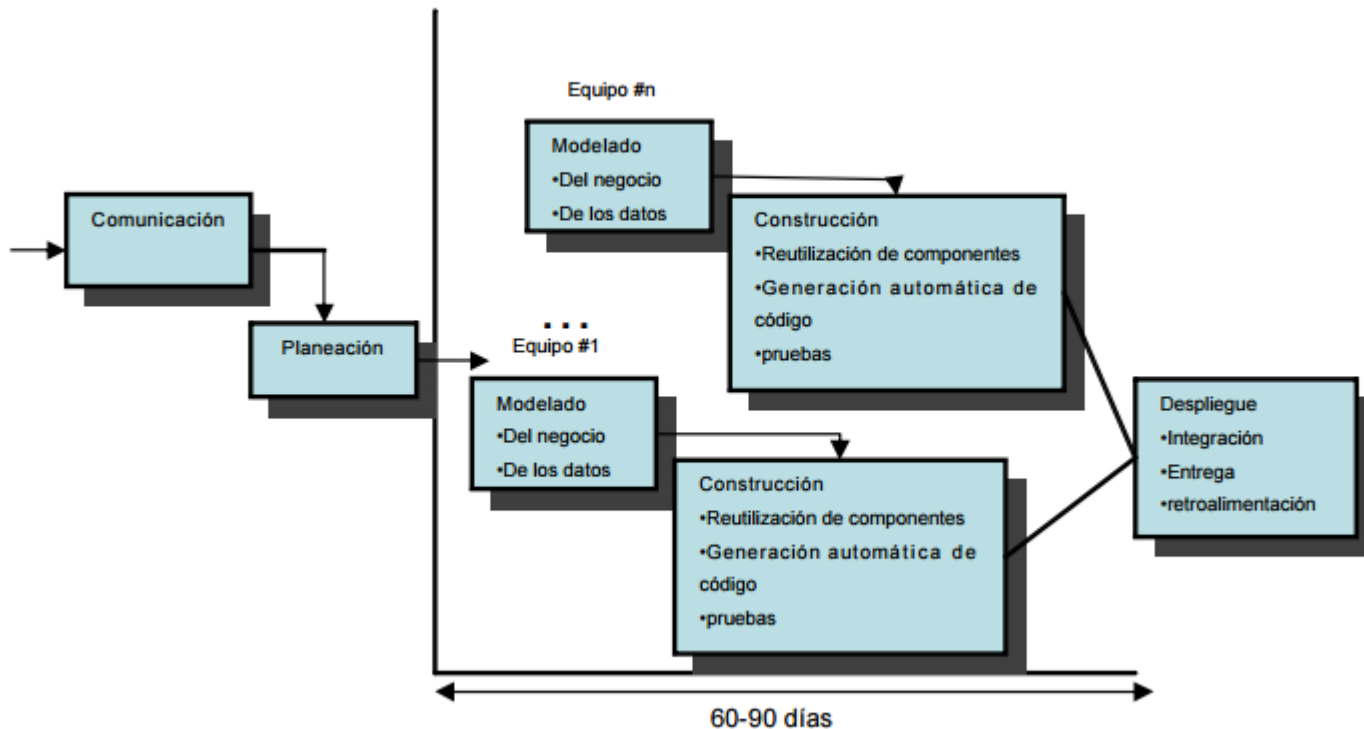
Desarrollado inicialmente por James Martín en 1980.

Comprende un desarrollo iterativo, construcción de prototipos y uso de utilidades CASE.

Acelera la creación de aplicaciones funcionales de 60 a 0 días.

# DRA (Desarrollo Rápido de Aplicaciones)

## Diagrama de Fases



Modelo de desarrollo rápido de aplicaciones (DRA)

Imagen obtenida de Méndez, 2006.



# DRA (Desarrollo Rápido de Aplicaciones)

## Descripción de las Fases

**Comunicación:** Inicio del proyecto, se realiza la recopilación de requerimientos con el cliente.

**Planeación:** Estimación de recurso y tiempos, se interpretan y validan los requisitos obtenidos.

**Modelado:** Se determina la estructura requerida para el negocio y para los datos. Algunos autores le llaman Modelado de Gestión y Modelado de Datos.



# DRA (Desarrollo Rápido de Aplicaciones)

## Descripción de las Fases

- 4. Construcción:** Elaboración del código. Algunos autores lo llaman Modelado de proceso y generación de aplicaciones. Incluye la reutilización de componentes, generación automática de código y pruebas.
- 5. Despliegue:** Incluye integración, pruebas de entrega y retroalimentación.



# DRA (Desarrollo Rápido de Aplicaciones)

## Ventajas



- Entrega temprana de un producto al cliente
- Mayor flexibilidad
- Ciclos de desarrollo más cortos con entregables
- Menor costo
- Menor codificación manual
- Compromiso del cliente con el sistema
- Aplicaciones fácilmente trasladables a otras plataformas

# DRA (Desarrollo Rápido de Aplicaciones)

## Desventajas

- Recomendables para proyectos medianos o pequeños
- Difícil de medir el avance de un proyecto
- Alto costo de herramientas integradas
- Menos precisión científica
- Los proyectos fallan si los desarrolladores y los clientes no se entienden
- Requieren de programadores con alta capacitación y habilidades.
- Requiere de una cantidad suficiente de personal para formar equipos eficientes.



# Modelos Evolutivos

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

- **Prototipo**
- **Espiral**
- **Concurrente**

# Prototipo

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Datos Generales



- Se utiliza cuando no se tiene detallados los requerimientos
- Se utiliza para dar al usuario una vista preliminar de cómo se encuentra el software
- Este modelo es básicamente prueba y error. Si al usuario no le gusta una parte del prototipo significa que la prueba falló por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho.

# Prototipo

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Diagrama de Fases

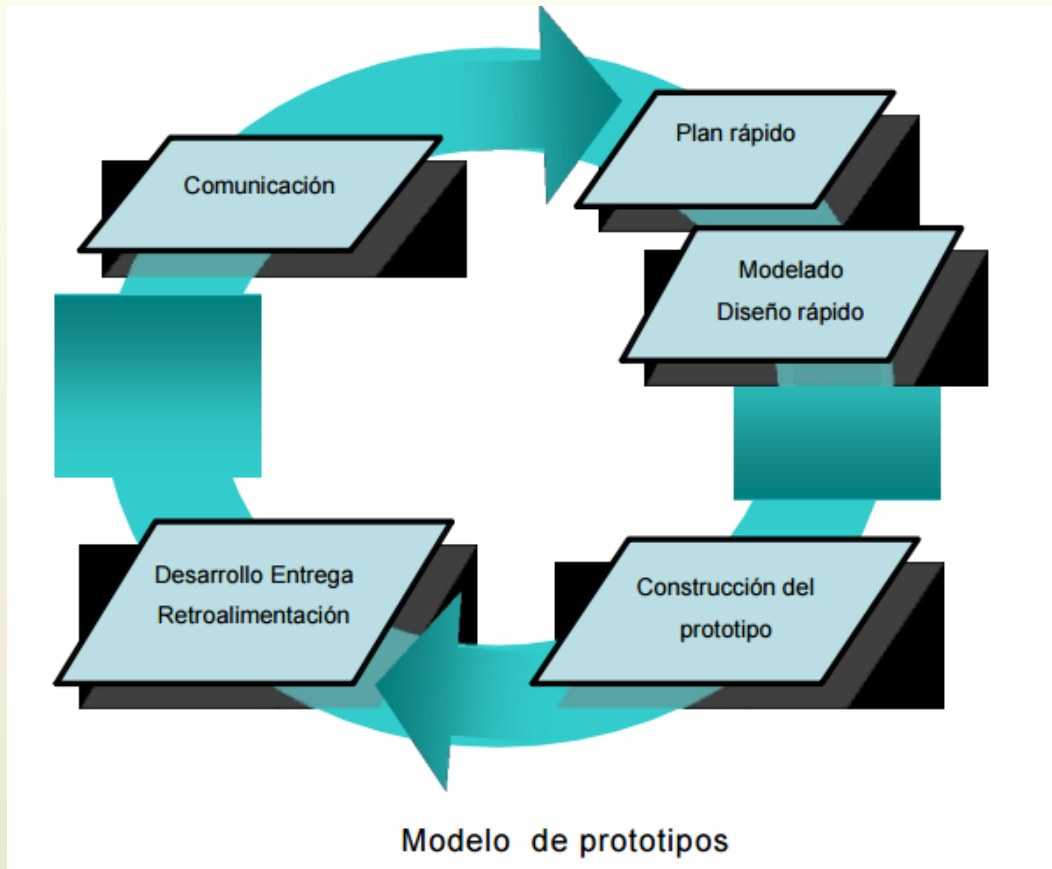


Imagen obtenida de Méndez, 2006.

# Prototipo

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE



## Descripción de las Fases

1. **Comunicación:** Inicio de la iteración, se realiza la recopilación de requerimientos con el cliente o las observaciones del cliente. Análisis de riesgo.
2. **Plan rápido:** Estimación de recurso y tiempos, se interpretan y validan los requisitos y observaciones obtenidas en la comunicación.
3. **Modelado y Diseño Rápido:** Se determina la estructura requerida para el software. Mejora del prototipo.



# Prototipo

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Descripción de las Fases

4. **Construcción del prototipo:** Elaboración del código.
5. **Desarrollo, entrega y retroalimentación:** Incluye integración, pruebas de entrega y retroalimentación.



# Prototipo

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Ventajas

- Permite el desarrollo a partir de requisitos poco claros.
- Reduce costos
- Útil al conocer los objetivos generales del software
- Ofrece un mejor enfoque cuando el responsable del desarrollo está inseguro de la eficacia.
- Más fácil de abordar con usuarios.
- El usuario participa activamente en la construcción del producto.
- Reduce riesgo de incertidumbre
- Proporciona al usuario un mayor conocimiento del sistema con una curva menor de aprendizaje.
- Facilita implementación final.



# Prototipo

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Desventajas

- El usuario no ve la diferencia entre prototipo y usuario final
- Los usuarios suelen enfocarse en aspectos “superficiales” del prototipo que los pueden dejar inconformes luego de verlos por primera vez.
- Requiere participación activa del usuario, quien no siempre tiene tiempo.
- La falta de experiencia que tienen muchos Analistas Funcionales en programación y en actividades de diseño de interfaces de usuario.

- 



# Espiral

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Datos Históricos



- Modelo atribuido a Boehm en 1988
- Es un modelo recomendado para proyectos cuya complejidad es importante y los costos que implica son más altos
- Integra características de modelo de cascada y el modelo de prototipos
- En cada giro se construye un nuevo modelo del sistema completo.
- Mejor modelo para el desarrollo de grandes sistemas.

# Espiral

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

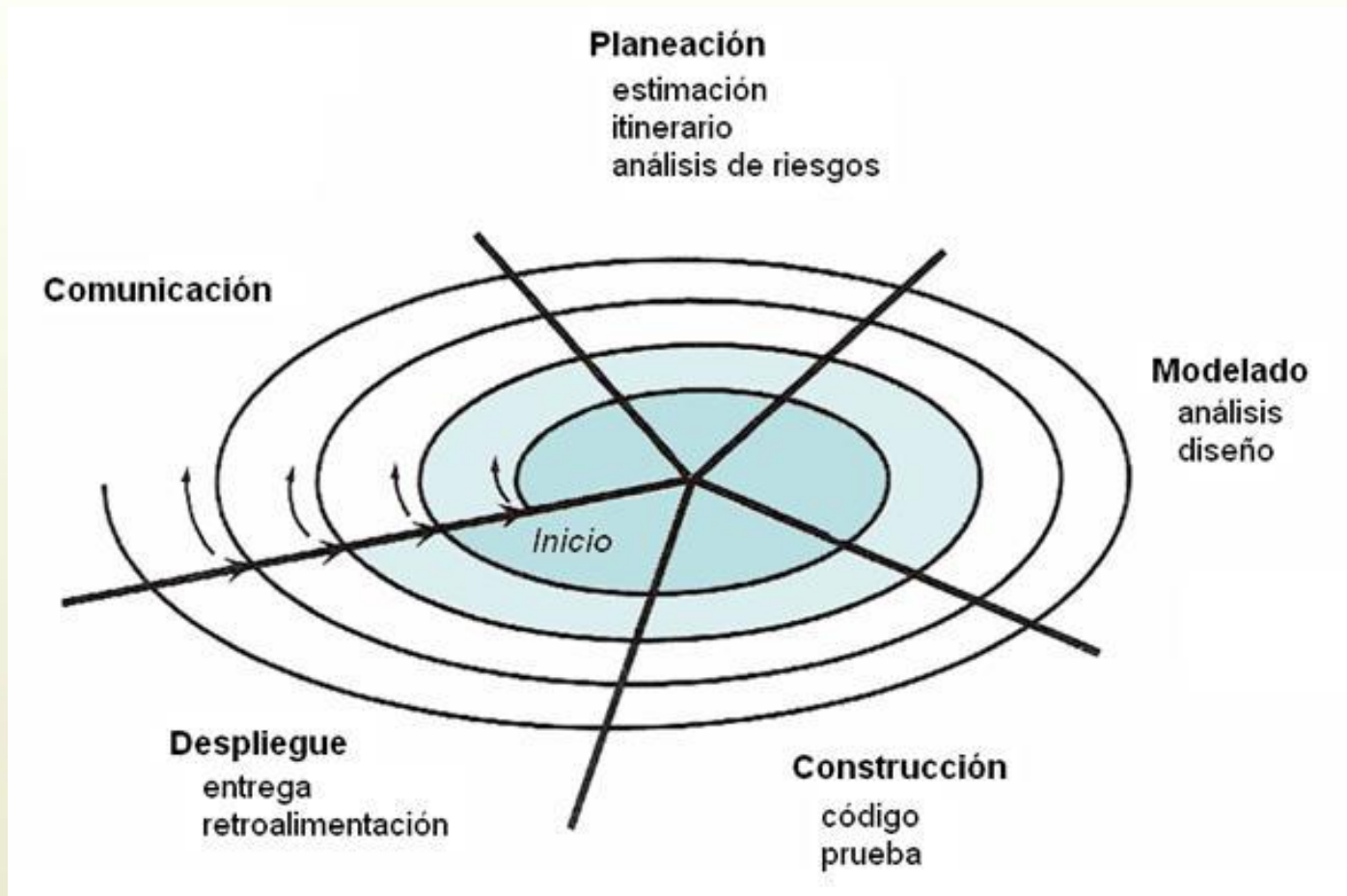


Imagen obtenida de: <http://www.aprenderaprogramar.com/foros/index.php?topic=2144.0>

# Espiral

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE



## Descripción de las Fases

**Comunicación:** Inicio de la iteración, se realiza la recopilación de requerimientos con el cliente o las observaciones del cliente. Análisis de riesgo.

**Planeación:** Estimación de recurso y tiempos, se interpretan y validan los requisitos y observaciones obtenidas en la comunicación.

**Modelado:** Se determina la estructura requerida para el software y/o mejora del producto. Incluye el análisis y diseño.

# Espiral

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Descripción de las Fases

4. **Construcción:** Elaboración del código. Y pruebas correspondientes.
5. **Despliegue:** Incluye integración, pruebas de entrega y retroalimentación.



# Espiral

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Ventajas

- Integra desarrollo con mantenimiento.
- Monitorea y controla los riesgos
- Permite a quien desarrolla aplicar el enfoque de prototipos en cualquier etapa de evolución del producto.
- Combina el ser metódico con el ser flexible e incrementa
- Puede adaptarse y aplicarse a lo largo de la vida del software
- Reduce los riesgos antes de que se conviertan en problemáticos
- Demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto.





# Espiral

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Desventajas

- Tiene una elevada complejidad
- Percepción de excesivos cambios por parte del cliente
- La ausencia de un plan fijo puede desorientar o causar mala impresión entre la dirección del proyecto y los empleados del mismo.
- Pérdida de tiempo al volver a iniciar
- Aplicable a proyectos de gran tamaño
- Utiliza mucho tiempo el desarrollo y finalización del proyecto
- Modelo costoso



# Concurrente

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

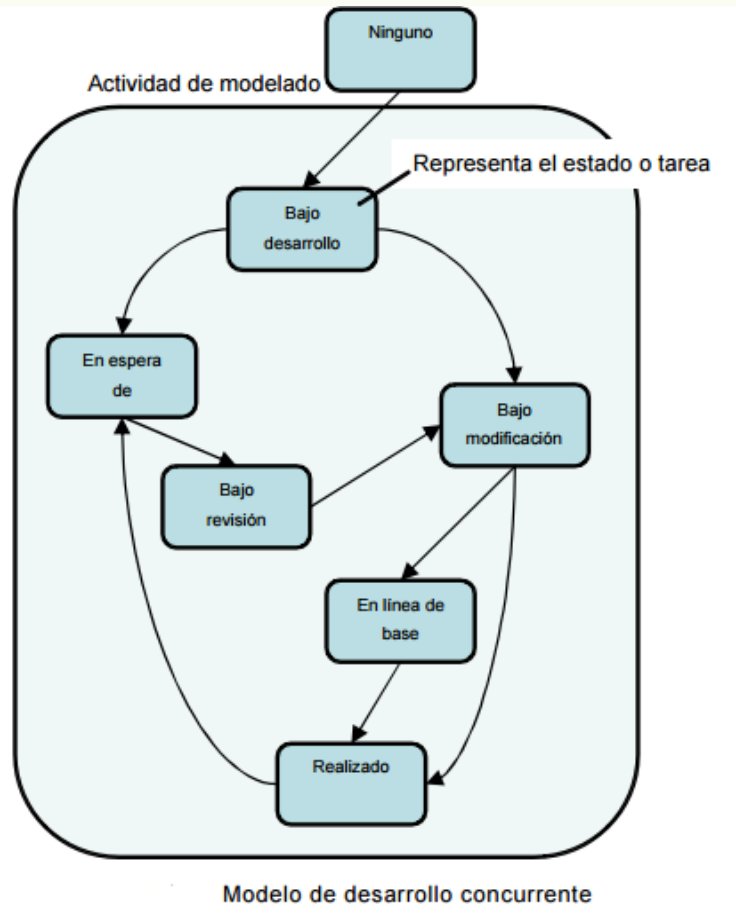
## Datos del modelo

- Atribuido a Davis Sitaram
- Se puede representar en forma de red o esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas.
- Define una serie de acontecimientos que disparan transiciones de estado a estado para cada actividad de ingeniería de software
- se puede expresar de manera esquematizada
- Las actividades llevan procesos concurrentes
- Aplicable para cliente soñador
- Dirigido por las necesidades del usuario
- Aplicable al modelo cliente/servidor



# Concurrente

## Diagrama de Fases



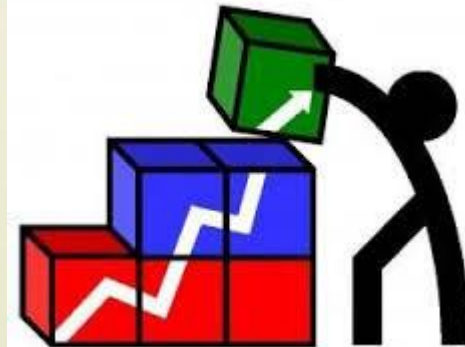
- La figura es una representación esquemática de una actividad(análisis).
- Todas las actividades existen concurrentemente, pero residen en estados diferentes.

Imagen obtenida de Méndez, 2006.

# Concurrente

## Descripción de las Fases

1. **Comunicación con el cliente** (no esta plasmada en la figura) y esta en estado de cambios en espera.
2. **Análisis** (tampoco se muestra en el esquema) significa que ya se ha hecho la comunicación con el cliente luego hace una transición al estado bajo desarrollo sin embargo si el cliente indica que se deben hacer cambios en requisitos , la actividad de análisis cambia del estado bajo desarrollo al estado cambios en espera.

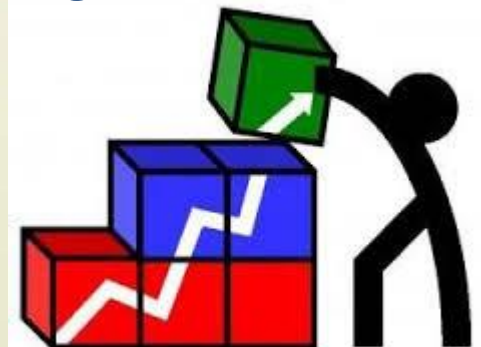


# Concurrente

## Descripción de las fases

3. **Bajo Desarrollo:** Aplicación de los cambios al proyecto.
4. **Cambios en espera:** cuando el cliente hace observaciones.

El modelo de proceso concurrente define una serie de acontecimientos que dispararan transiciones de estado a estado para cada una de las actividades de la ingeniería del software.



# Concurrente

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

## Ventajas



- Eficiencia en tiempo
- Excelente para proyectos en los que se conforman grupos de trabajo independientes
- Utilizado en todo tipo de desarrollo de software
- Proporciona una visión certera del estado actual del proyecto
- Se realizan varias actividades al mismo tiempo

# Concurrente

## Desventajas

- Durante las primeras etapas del diseño, no se contempla una inconsistencia del modelo de análisis
- Si no se dan las condiciones señaladas no es aplicable.
- Si no existen grupos de trabajo no se puede trabajar en este método



# Conclusiones

CONTROLLED COLOR RANGE  
COMPLETE RGB GAMMA  
CMYK TOLERANCE

- Cada uno de los modelos presentados tiene sus pros y sus contras.
- Al realizar un desarrollo se deben considerar las características del mismo para elegir el mejor modelo
- Se deben considerar las características del equipo de desarrollo para aplicar uno u otro modelo.
- Los modelos evolutivos son actualmente los más utilizados





# Referencias

1. López, R. O., & Ramiro, L. V. (2011). Ingeniería del software. Centro de Estudios Financieros.
2. Méndez Nava, E. M., & RAMÓN, G. (2006). Modelo de evaluación de metodologías para el desarrollo de software. *Caracas, Venezuela*.
3. Pressman, R. S., & Troya, J. M. (2010). *Ingeniería del software*. McGraw Hill
4. Sommerville, I., & Galipienso, M. I. A. (2005). Ingeniería del software. Pearson Educación.

## Referencias Electrónicas

1. Modelo Lineal Secuencial (Cascada). Disponible en: <https://prezi.com/q4giwjxltgj/modelo-lineal-secuencial/> Fecha de Consulta: 4 de octubre de 2016
2. Paradigmas de Ingeniería de software. Disponible en: [http://148.204.211.134/polilibros/Portal/Polilibros/P\\_proceso/ANALISIS\\_Y\\_DISENO\\_DE\\_SISTEMAS/IngenieriaDeSoftware/CIS/UNIDAD%20I/1.5.htm](http://148.204.211.134/polilibros/Portal/Polilibros/P_proceso/ANALISIS_Y_DISENO_DE_SISTEMAS/IngenieriaDeSoftware/CIS/UNIDAD%20I/1.5.htm). Fecha de consulta 4 de octubre de 2016.
3. Procesos de Software. Disponible en : <http://procesosdesoftware.blogspot.mx/>. Fecha de consulta 4 de octubre de 2016.
4. Ingeniería de Software. Disponible en: <http://ingenieria.uatx.mx/labastida/files/2013/01/Modelo-Incremental.pdf>. Fecha de consulta: 4 de octubre de 2016

