Diseño de Sistemas de Información

LUIS EDIN IBARGUEN MOSQUERA

TALLER N°. 1

ING. CARLOS ALBERTO LONDOÑO LOAIZA

CORPORACIÓN DE ESTUDIOS

TECNOLÓGICOS

DEL NORTE DEL VALLE

PROGRAMA

CARTAGO

2017

Contenido

[Tabla de Ilustraciones 2](#_Toc490929124)

[Taller No. 1 3](#_Toc490929125)

[EL PROCESO DEL SOFTWARE 3](#_Toc490929126)

[MODELOS DEL PROCESO DEL SOFTWARE 4](#_Toc490929127)

[MODELO CASCADA 4](#_Toc490929128)

[MODELO ESPIRAL 7](#_Toc490929129)

[MODELO INCREMENTAL 10](#_Toc490929130)

[MODELO ITERATIVO 13](#_Toc490929131)

[CUADRO COMPARATIVO DE LOS MODELOS DE PROCESO DE SOFTWARE 15](#_Toc490929132)

[Bibliografía 17](#_Toc490929133)

# Tabla de Ilustraciones

[1 Diagrama Cascada 6](#_Toc490928895)

[2 Diagrama Espiral 8](#_Toc490928896)

[3 Diagrama Incremental 12](#_Toc490928897)

[4 Diagrama Iterativo 15](#_Toc490928898)

# Taller No. 1

# EL PROCESO DEL SOFTWARE

Proceso: conjunto de actividades ejecutadas en una secuencia de instrucciones.

Este permite que las personas que elaboran el trabajo (equipo de Software) busquen y elijan el conjunto adecuado de acciones y tareas para desarrollarlo, buscando entregar el Software de forma rápida y con una excelente calidad para satisfacer a los clientes. Este proceso como tal requiere un conjunto de actividades llamadas actividades estructurales usadas a través de todo el proceso del Software y son 5:

**Comunicación:** con el cliente y los participantes para definir objetivos del proyecto, requerimientos y funciones del Software.

**Planeación:** se elabora un mapa guía llamado Plan de Proyecto de Software que define tareas por realizar.

**Modelado:** se crean modelos con el fin de mejorar los requerimientos y el diseño.

**Construcción:** se genera un código (manual o automatizado) y se elaboran pruebas para descubrir errores en éste.

**Despliegue:** terminado el Software se entrega al cliente para que lo ensaye y evalúe.

Estas actividades son complementadas por las actividades sombrilla o de apoyo las cuales ayudan a controlar el avance, la calidad, el cambio y el riesgo del Software, las actividades sombrilla son:

Seguimiento y control del Proyecto de Software.

Administración del riesgo.

Aseguramiento de la calidad del Software.

Revisiones técnicas.

Medición.

Administración de la configuración del Software

Administración de la reutilización.

Preparación y producción del producto del trabajo.

Estas también ocurren a lo largo del proceso de Software y se centran sobre todo en la administración, el seguimiento y el control del proyecto.

# MODELOS DEL PROCESO DEL SOFTWARE

Es una representación abstracta de un proceso del software que pueden utilizar para explicar diferentes enfoques para el desarrollo del software. Los modelos del proceso son: Modelo Cascada, Modelo Incremental, Desarrollo Iterativo, Desarrollo Evolutivo o Espiral.

## MODELO CASCADA

CARACTERISTICAS:

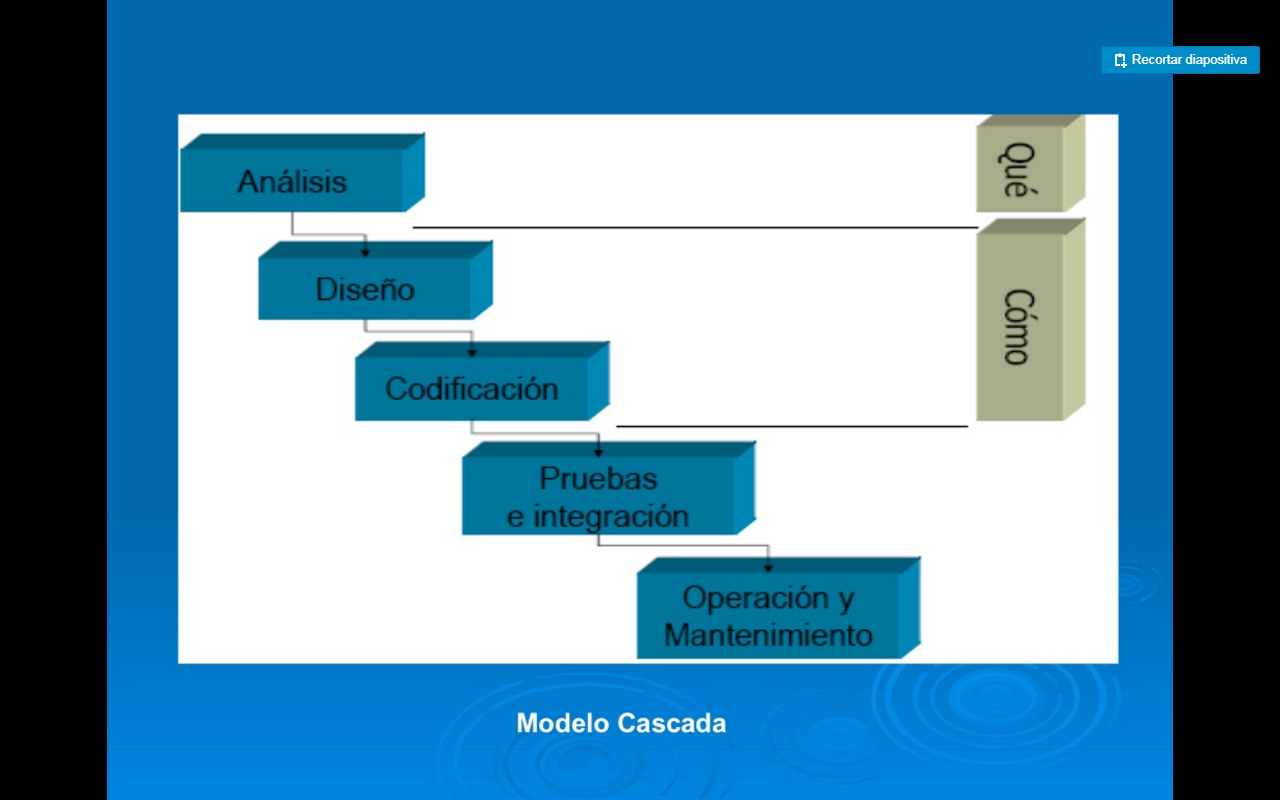
Es el más utilizado.

Es una visión del proceso de desarrollo de software como una sucesión de etapas que producen productos intermedios.

Para que el proyecto tenga éxito deben desarrollarse todas las fases.

Las fases continúan hasta que los objetivos se han cumplido.

Si se cambia el orden de las fases, el producto final será de inferior calidad.



1 Diagrama Cascada

Planificación: Realiza un estudio de factibilidad del software así como contemplar los posibles costos que pueden surgir mediante su implementación.

Análisis y Diseño de Requerimientos: Involucra la identificación de las características que nos guían para determinar las funcionalidades del software de acuerdo al medio donde se pretende implementar, es muy importante notar que trata de responder a las preguntas ¿Quienes intervienen en el uso del Software?, ¿Qué restricciones tendrá el software?

Diseño: Se identifica y describe las abstracciones del software y cumplir con los requerimientos plasmando todas esas características en un diseño que permite visualizar y contemplar adicionalmente situaciones no previstas.

Implementación: Realizar las pruebas pertinentes y verificar que se cumplen con las características identificadas.

Operación y Mantenimiento: Se instala dentro del ambiente, dependerá que pasará a partir de ahí, ya que esta etapa aún puede considerar nuevamente la existencia de características que no han sido contempladas y/o características innecesarias, implicando la modificación del software para la adaptación de estas anomalías.

Crecimiento y cambio: Se evalúa el software de modo que se determina si se puede emplear dentro de la nueva tecnología no afectando la integridad del mismo, de modo que si no es posible que exista una adaptación a lo nuevo, el proceso de diseño del software nuevamente se repite desde el principio.

**DESVENTAJAS**

Se tarda mucho tiempo en pasar por todo el.

El mantenimiento se realiza en el código fuente.

Las revisiones de proyectos de gran complejidad son muy difíciles.

**LIMITACIONES**

No se permiten las iteraciones.

Los requisitos se congelan al principio del proyecto.

No existe un proyecto “enseñable” hasta el final del proyecto.

## MODELO ESPIRAL



2 Diagrama Espiral

Comunicación con el cliente: las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.

Planificación: las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.

Análisis de riesgos: las tareas requeridas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.

Ingeniería: las tareas requeridas para construir una o más representaciones dela aplicación.

Construcción y adaptación: las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

Evaluación el cliente: las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementación durante la etapa desinstalación

**CARACTERISTICAS**

Incorpora objetivos de calidad y gestión de riesgos.

Elimina errores y alternativas.

Permite iteraciones, vuelta atrás y finalizaciones rápidas

Cada ciclo empieza identificando:

-Los objetivos de la porción correspondiente.

-Las alternativas

-Restricciones

Cada ciclo se completa con una revisión que incluye todo el ciclo anterior y el plan para el siguiente.

**VENTAJAS**

Utiliza las fases de modelos tradicionales. Se centra en la eliminación de errores y alternativas poco atractivas.

Su orientación a detectar y prevenir el riesgo evita muchas dificultades.

El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.

Como el software evoluciona a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los nivele evolutivos.

El modelo en espiral permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.

El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemas.

En la utilización de grandes sistemas a doblado la productividad.

**DESVENTAJAS**

Complicado: Consume muchos recursos

Las etapas y sus E/S no están claramente definidas.

Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.

Debido a su elevada complejidad no se aconseja utilizarlo en pequeños sistemas.

Genera mucho tiempo en el desarrollo del sistema

Modelo costoso

Requiere experiencia en la identificación de riesgos.

## MODELO INCREMENTAL

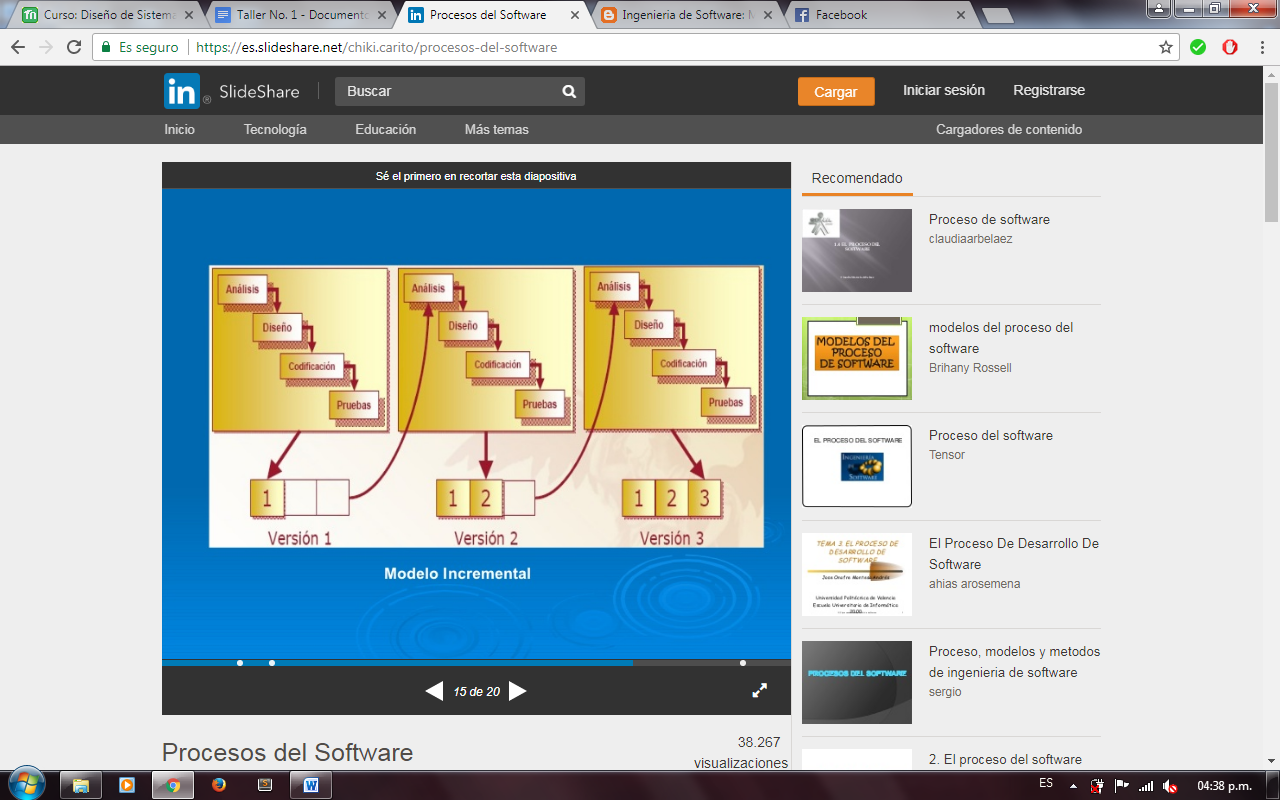
Es una repetición de varios ciclos de vida en cascada.

Al final de cada ciclo se entrega una versión parcial del software incrementada con cierta funcionalidad nueva respecto a las entregas anteriores.

Los ciclos se repiten hasta obtener un producto completo.

Los usuarios disponen antes del software, aunque no sea completo, por lo que pueden sugerir mejoras.

Se suele aplicar a desarrollos de gran tamaño.



3 Diagrama Incremental

**CARACTERÍSTICAS**

- Se evitan proyectos largos y se entrega “algo de valor” a los usuarios con cierta frecuencia.

- El usuario se involucre más.

- Difícil de evaluar el costo total.

- Difícil de aplicar a los sistemas transaccionales que tienden a ser integrados y a operar como un todo.

- Requiere gestores experimentados.

- Los errores en los requisitos se detectan tarde.

- El resultado puede ser muy positivo.

**VENTAJAS:**

- Con un paradigma incremental se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.

- También provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del Software.

- El modelo proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.

- Permite entregar al cliente un producto más rápido en comparación del modelo de cascada.

- Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos.

- Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico.

**DESVENTAJAS:**

- El modelo Incremental no es recomendable para casos de sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido, y/o de alto índice de riesgos.

- Requiere de mucha planeación, tanto administrativa como técnica.

- Requiere de metas claras para conocer el estado del proyecto.

## MODELO ITERATIVO

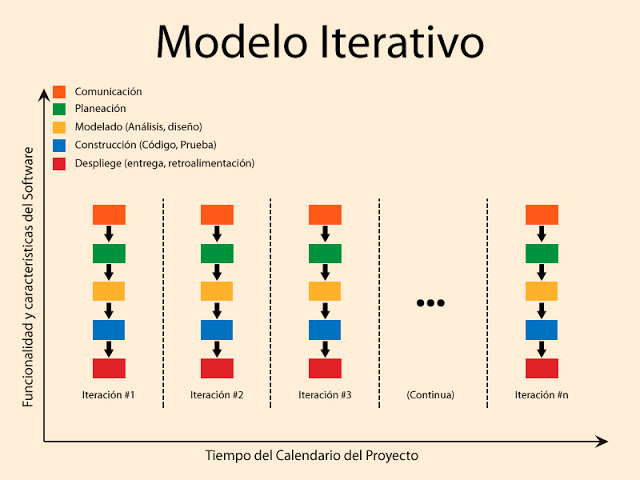
Es también una repetición de varios ciclos de vida en cascada.

Al final de cada ciclo se entrega una versión completa del software mejorada respecto a la anterior. Los ciclos se repiten hasta obtener un producto satisfactorio.

Los usuarios deben evaluar el producto en cada iteración y proponer mejoras.

Se suele aplicar en desarrollos en los que los requisitos no están claros, las primeras versiones pueden ser prototipos que se desechan posteriormente.

Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente.



4 Diagrama Iterativo

**Ventajas**

Una de las principales ventajas que ofrece este modelo es que no hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir refinando en cada una de las iteraciones.

Igual que otros modelos similares tiene las ventajas propias de realizar el desarrollo en pequeños ciclos, lo que permite gestionar mejor los riesgos, gestionar mejor las entregas…

**Inconvenientes**

La primera de las ventajas que ofrece este modelo, el no ser necesario tener los requisitos definidos desde el principio, puede verse también como un inconveniente ya que pueden surgir problemas relacionados con la arquitectura.

# CUADRO COMPARATIVO DE LOS MODELOS DE PROCESO DE SOFTWARE

|  |  |  |
| --- | --- | --- |
| Modelos de proceso de Software. | Ventajas | Desventajas |
| Modelo de cascada. | 1. Modelo y planificación fácil y sencilla.  2. Sus fases son conocidas por los desarrolladores.  3. Los usuarios lo pueden comprender fácilmente.  4. El producto posee alta calidad  5. Permite trabajar con personas de pocos estudios  6. Sigue los pasos intuitivos necesarios a la hora de desarrollar el software.  7. Es un modelo secuencial ya que requiere retroalimentación  8. Se debe de planear y programar las actividades del proceso antes de trabajar | 1. Se tarda mucho tiempo en pasar por todo el ciclo  2. El mantenimiento se realiza en el código fuente  3. Las revisiones de proyectos de gran complejidad son muy difíciles  4. Impone una estructura de gestión de proyectos  5. Alto riesgo en sistemas nuevos debido a problemas en las especificaciones y en el diseño.  6. Difícil para el cliente establecer explícitamente al principio todos los requisitos. |
| Modelo de espiral. | 1. Puede adaptarse y aplicarse a lo largo de la vida del software de computadora.  2. Es un enfoque realista del desarrollo de sistemas y de software a gran escala.  3. Mejor reaccionan ante riesgos en cada uno de los niveles evolutivos.  4. Utiliza la construcción de prototipos como mecanismo de reducción de riesgos.  5. Permite aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.  6. Si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos | 1. Puede resultar difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.  2. Requiere una considerable habilidad para la evaluación del riesgo.  3. Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas.  4. Este sistema es muy utilizado en proyectos largos. Y que necesitan constantes cambios.  5. Al no ser detectados y catalogados los riesgos a tiempo, puede llevar al fracaso. |
| Modelo incremental. | 1. se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.  2. proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.  3. más rápido en comparación del modelo de cascada.  4. Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos. | 1. El modelo Incremental no es recomendable para casos de sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido, y/o de alto índice de riesgos.  2. Requiere de mucha planeación, tanto administrativa como técnica.  3. Requiere de metas claras para conocer el estado del proyecto. |
| Modelo Iterativo | 1. En el desarrollo de este modelo se da la retroalimentación muy temprano a los usuarios.  2. Permite separar la complejidad del proyecto, gracias a su desarrollo por parte de cada iteración o bloque.  3. El producto es consistente y puntual en el desarrollo.  Los productos desarrollados con este modelo tienen una menor probabilidad de fallar.  4. Se obtiene un aprendizaje en cada iteración que es aplicado en el desarrollo del producto y aumenta las experiencias para próximos proyectos. | 1. Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que simplemente no estarán dispuestos a invertir el tiempo necesario.  2. Infunde responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente, requiriendo de profesionales sobre el promedio.  3. Sufre fuertes penalizaciones en proyectos en los cuales los requerimientos están previamente definidos, o para proyectos "todo/nada" en los cuales se requiere que se completen en un 100% el producto para ser implementado (por ejemplo, licitaciones) |

# Bibliografía

(s.f.). Obtenido de http://www.mitecnologico.com/Main/ModeloBasadoEnComponentesDise%F1oDeSistemas

(s.f.). Obtenido de http://www.slideshare.net/chiki.carito/procesos-del-software

(s.f.). Obtenido de www.e-market.cl/dir/umayor/ingsw/06-01\_vesp/espiral.ppt

(s.f.). Obtenido de www.dsic.upv.es/asignaturas/.../lsi/.../IntroduccionProcesoSW.doc

(s.f.). Obtenido de http://www.mitecnologico.com/Main/ModelosDeProcesoDeSoftware

(s.f.). Obtenido de http://html.rincondelvago.com/modelos-de-procesos-de-software.html

(s.f.). Obtenido de http://www.slideshare.net/JoanFernandoChipia/modelos-basados-en-prototipos

(s.f.). Obtenido de http://148.202.148.5/cursos/cc321/fundamentos/unidad1/espiral.htm

(s.f.). Obtenido de http://scruz334.blogspot.es/1193793960/

(s.f.). Obtenido de http://es.wikipedia.org/wiki/Ingenier%C3%ADa\_de\_software

(s.f.). Obtenido de http://www.buenastareas.com/ensayos/Ingenieria-De-Software-Basada-En-Componentes/232694.html

(s.f.). Obtenido de http://es.wikipedia.org/wiki/Software#Modelos\_evolutivos

(s.f.). Obtenido de http://ingenieraupoliana.blogspot.com.co/2010/10/modelo-incremental.html

(s.f.). Obtenido de http://modeloespiral.blogspot.com.co/

(s.f.). Obtenido de https://es.slideshare.net/claudiaarbelaez/proceso-de-software

(s.f.). Obtenido de http://www.paginaspersonales.unam.mx/files/1065/Modelos\_de\_procesos\_del\_software.pdf

(s.f.). Obtenido de http://es.slideshare.net/toryneutral/metodos-formales

(s.f.). Obtenido de http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P\_proceso/ANALISIS\_Y\_DISEnO\_DE\_SISTEMAS/IngenieriaDeSoftware/CIS/UNIDAD%20I/1.5.htm

(s.f.). Obtenido de http://cetis733cequpo008.blogspot.com/2006/10/modelo-dra-el-desarrollo-rapido-de.html