

ADOO. PROYECTOS DE SISTEMAS DE INFORMACION - IT:

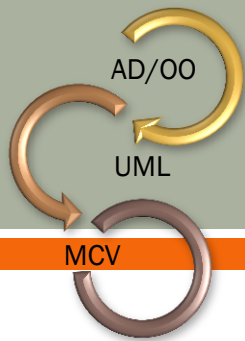
ESTRATEGIAS METODOLÓGICAS PARA EL DESARROLLO DE LOS SISTEMAS DE INFORMACIÓN AUTOMATIZADO.

En este documento:

Analizaremos los pasos a seguir para hacer que un proyecto de desarrollo de un Sistema de Información automatizado tenga tanto éxito como sea posible.

Mg. Jorge Medianero Acosta
Docente FIIS / UNI

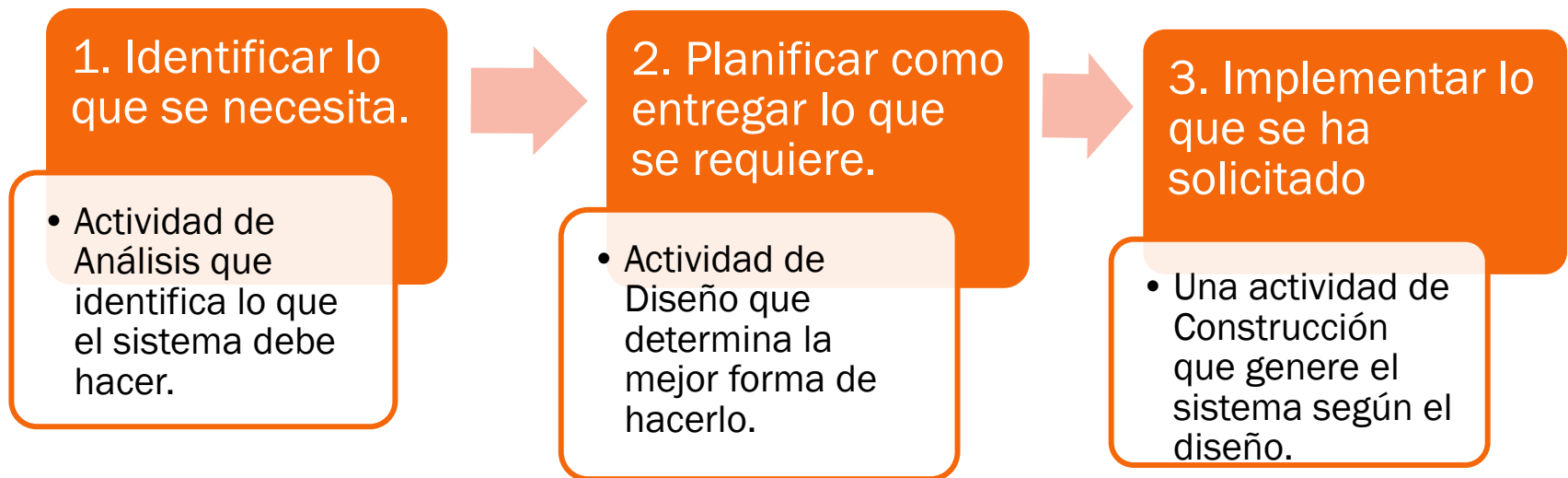




ADOO.

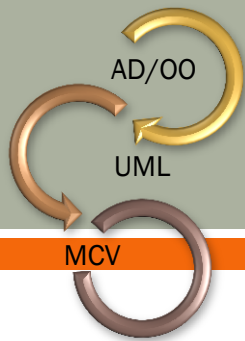
Construcción de Sistemas de Información basado en Ordenadores

Desde una perspectiva general, la construcción de SI basado en PC tiene un objetivo: La resolución de problemas. Para lograr este objetivo, el proceso de desarrollo de los SI pueden dividirse en tres tareas principales.



NOTA: Tradicionalmente se han asignado diferentes nombres a las fases que abarcan estas entidades, pero los temas centrales son siempre lo mismo.



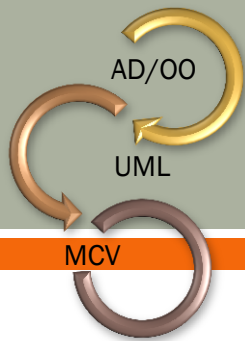


AD00.

Construcción de Sistemas de Información basado en Ordenadores

- Una fuente adicional de ayuda de la que dispone el diseñador de software es el uso de los ordenadores. El Desarrollo de software es una actividad que como mucha otras se beneficia del empleo de la PC.
- El apoyo implícito para el desarrollo de software no se hizo realidad en lo que respecta a los costes y tecnología, hasta mediados de la década de 1980.
- En la actualidad, se acepta que el apoyo informático es un componente importante de cualquier proyecto de desarrollo (Allen, 1991).



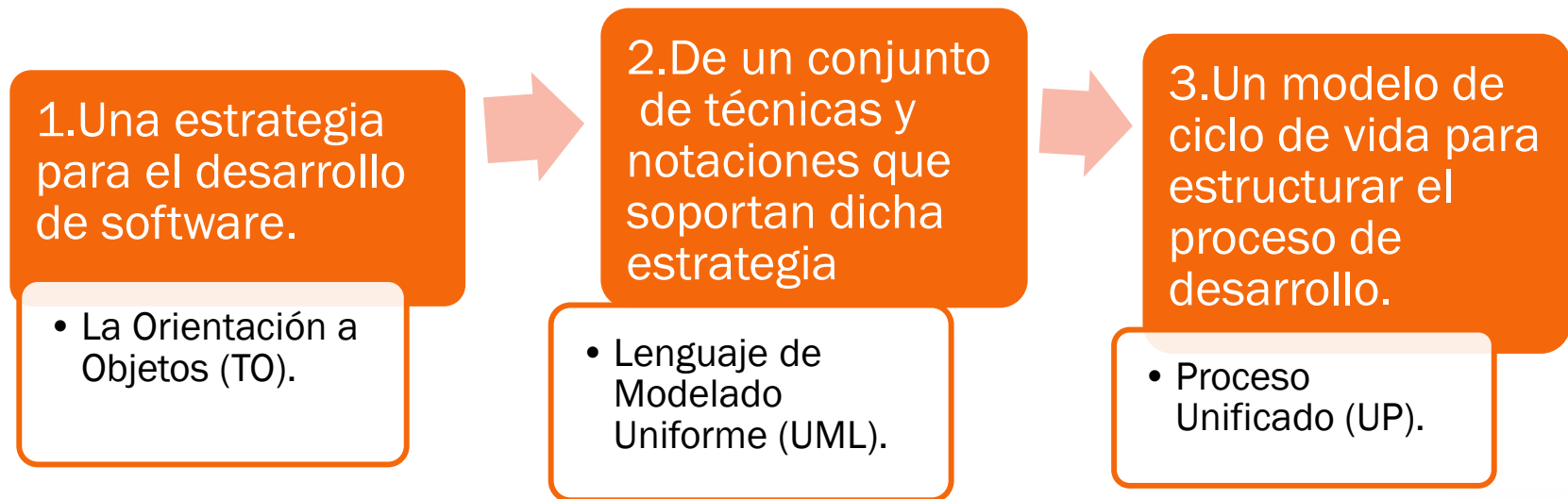


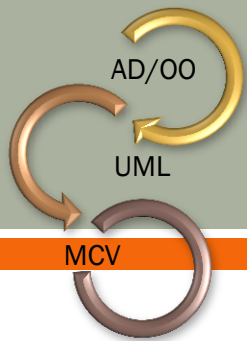
ADOO.

IMPORTANCIA DE LAS ESTRATEGIAS METODOLOGICAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACION – IT.

Una consecuencia importante de subdividir el proceso de desarrollo de Sistemas de Información es que se pueden identificar las estrategias metodológicas necesarias en cada fase. Uno de los factores que mas puede influir en el éxito o fracaso de los Sistemas a construir, es la estrategia de desarrollo que se adopte.

Una metodología consta de las siguientes estrategias de desarrollo de SI:





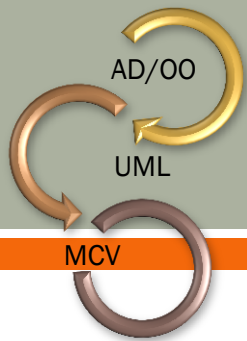
AD00.

Estrategia 1: La Orientación a Objetos (OO)

Tecnología OO.

- La estrategia Orientada a objetos proporciona un mecanismo para traducir los problemas del mundo real en abstracciones que el **software** puede plasmar eficazmente.
- La estrategia Orientada a objetos proporciona un mecanismo para traducir los problemas del mundo real en abstracciones que el **software** puede plasmar eficazmente.
- Además, la OO proporciona **estructuras conceptuales** que permiten construir **modelos de sistemas** de información **complejos**.
- Es una estrategia que permite transformar el desarrollo de un sistema grande y complejo en la elaboración de un conjunto de subsistemas mas simples.



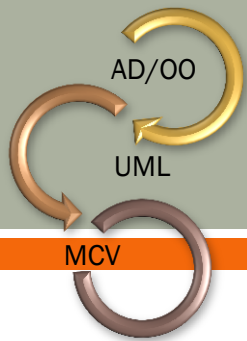


ADOO. Estrategia OO: Ventajas.

Ventajas de la OO: (Tecnología de objetos).

- La OO dispone de estructuras conceptuales que apoyan la **modularización** (subdivisión en subsistemas).
- La OO aspira a proporcionar un mecanismo que facilite la **reutilización** de código de programación, del diseño, y de los modelos de análisis.



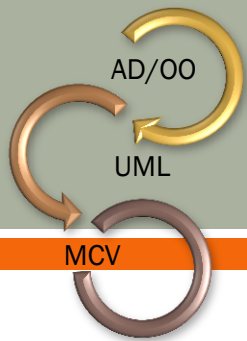


ADOO. Estrategia OO: Limitaciones.

Desventajas de la OO: (Tecnología de objetos).

- Algunas aplicaciones no se encuentran idealmente adaptadas al desarrollo OO.
- El primer tipo incluye sistemas que están fuertemente orientadas a base de datos (estructura de datos basada en registros, centrados en el almacenamiento y recuperación de datos: RDBMS). Los RDBMS comerciales son una tecnología muy madura que organiza sus datos de acuerdo a principios matemáticos muy definidos (Teoría relacional).
- Sin embargo, los R-DBMS están limitados en su capacidad para almacenar y recuperar ciertos tipos de estructuras complejas de datos, tales como datos multimedia (mapas geodésicos), pero que si se ajustan de manera ideal al desarrollo OO y por tanto a OO-DBMS.





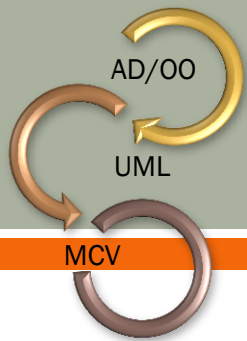
ADOO.

Estrategia 2: Técnicas y Notación de Modelado UML

Técnicas de Modelado UML.

- El objetivo principal del análisis y del diseño de proyectos de desarrollo se centra en los modelos (es decir, tener un sistema que funcione).
- Se pueden modelar muchos tipos de cosas. Los Ing. Civiles elaboran modelos de puentes, los urbanistas modelan el transporte, un científico modela el impacto ambiental de residuos mineros,
- Un analista de sistemas elabora un modelo de información para una empresa.





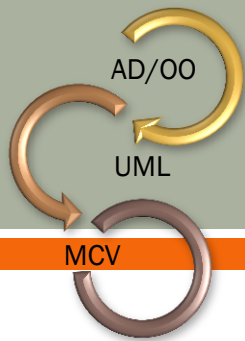
AD/OO.

Estrategia 2: Técnicas y Notación de Modelado UML

Que es un modelo? Como sucede con los mapas, los modelos representan algo:

- Un modelo es mas rápido y fácil de construir.
- Un modelo es una abstracción: Un modelo puede representar elementos reales o imaginarios de cualquier dominio.
- Un modelo puede evolucionar a medida que asimilamos el problema de estudio.
- Podemos elegir que detalles queremos representar en un modelo y cuales ignorar.





ADOO.

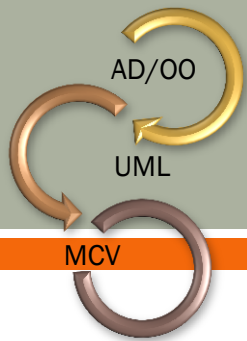
Estrategia 2: Técnicas y Notación de Modelado UML

Que es un diagrama? Los diseñadores utilizan diagramas para construir modelos de los sistemas de la misma forma que los arquitectos utilizan símbolos y diagramas para modelar edificios.

El ADOO, utiliza con frecuencia a los modelos basados en diagramas para:

- Comunicar ideas.
- Comprender las estructuras (artefactos) y sus relaciones.





AD/OO.

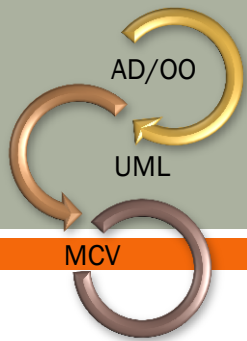
Estrategia 2: Técnicas y Notación de Modelado UML

Modelos UML y Sistemas Empresariales?

En el AD/OO, los modelos pueden ser sistemas empresariales ya existentes o pueden ser de nuevos sistemas informáticos.

- Si un sistema es muy sencillo, podrá ser modelado utilizando un único diagrama y escribiendo descripciones textuales.
- La mayoría de los sistemas son mas complejos y pueden exigir la representación de numerosos diagramas para capturar toda su complejidad.





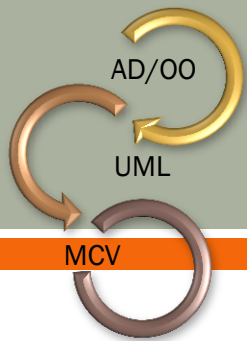
ADOO.

Estrategia 2: OMG & la Especificación UML

Especificación UML. UML/OMG, provee un conjunto de técnicas y notaciones descritas en la especificación UML. (Típicos diagramas para el ADOO).

- UML emplea formas abstractas para representar objetos y acciones del mundo real.
- La elección de las formas a usar viene determinada por un conjunto de reglas según el tipo particular de diagrama.
- En UML estas reglas están definidas en la especificación 2.0 del lenguaje de Modelado Unificado OMG (OMG Language Specification 2.0, OMG 2004c).
- Las normas son importantes porque estimulan la comunicación de la misma forma que lo hace un idioma.





AD/OO.

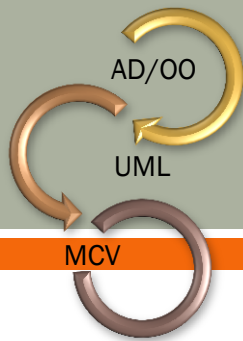
Estrategia 2: OMG & la Especificación UML

Especificación UML. Para los diseñadores de modelado, las técnicas deberán promover el uso de algunas reglas fundamentales como:

- **Simplicidad de Representación.** Solo tiene que mostrarse lo que se necesita mostrar.
- **Representación Jerárquica.** Descomponiendo el sistema y mostrando mas detalles relevantes a niveles inferiores (Consistencia interna y Plenitud).

UML es el soporte del modelado basado en objetos. El AD/OO, usa UML como una estrategia de modelado ya que se ha convertido en el estándar industrial para el desarrollo de Sistemas de Información





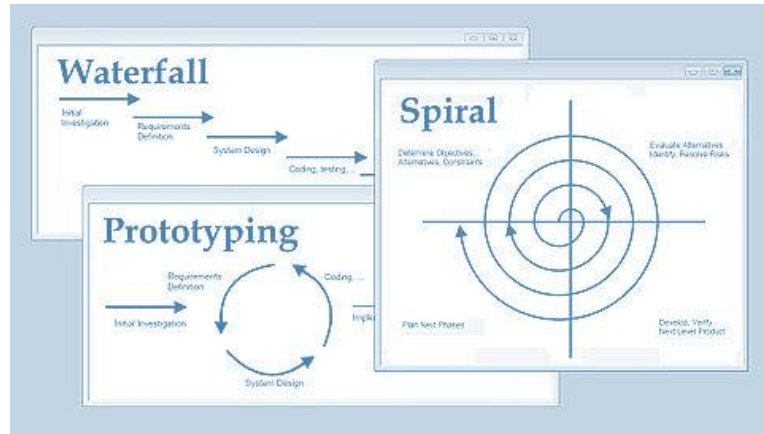
AD/OO.

Estrategia 3: Modelos de Ciclos de Vida (MCV)

LCM - Life Cycle Model.

Modelos de Ciclo de Vida de los proyectos.

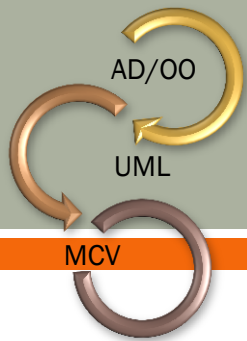
Subdividir el proceso de desarrollo de software produce lo que se conoce como un Modelo de Ciclo de Vida (LCM: Life Cycle Model).



LCM: Tipos de modelos mas utilizados.

- 1. Ciclo de vida Tradicional** (TLC 1960). Este modelo también se conoce como el Ciclo de Vida en Cascada.
- 2. Modelo en espiral** de Boehm (1988). Es un modelo adaptable de desarrollo incremental de apoyo.
- 3. Proceso unificado de Desarrollo de Software** (USDP Jacobson et. al. 1999). Este modelo refleja el énfasis actual existente sobre los ciclos de vida Iterativos e Incremental.





ADOO.

Estrategia 3: Modelos de Ciclos de Vida (MCV)

LCM Life Cycle Model: Ciclo de Vida de los proyectos de Software.

Definición. El proceso que se sigue para construir, entregar y hacer evolucionar el software desde la concepción del proyecto de desarrollo hasta la entrega y retiro del sistema.

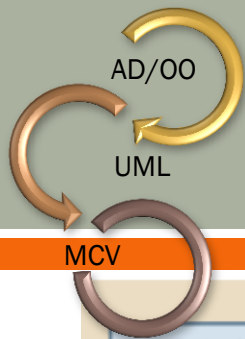
Premisas del modelo de proceso.

- ¿Qué debemos hacer a continuación ?
- ¿Por cuánto tiempo debemos hacerlo

Objetivos.

- 1.- Boehm: Determinar el orden de las etapas de desarrollo de software.
- 2.- Establecer el criterio de transición para progresar de una etapa a la siguiente:
 - Criterio para determinar la finalización.
 - Criterio para comenzar y elegir la siguiente

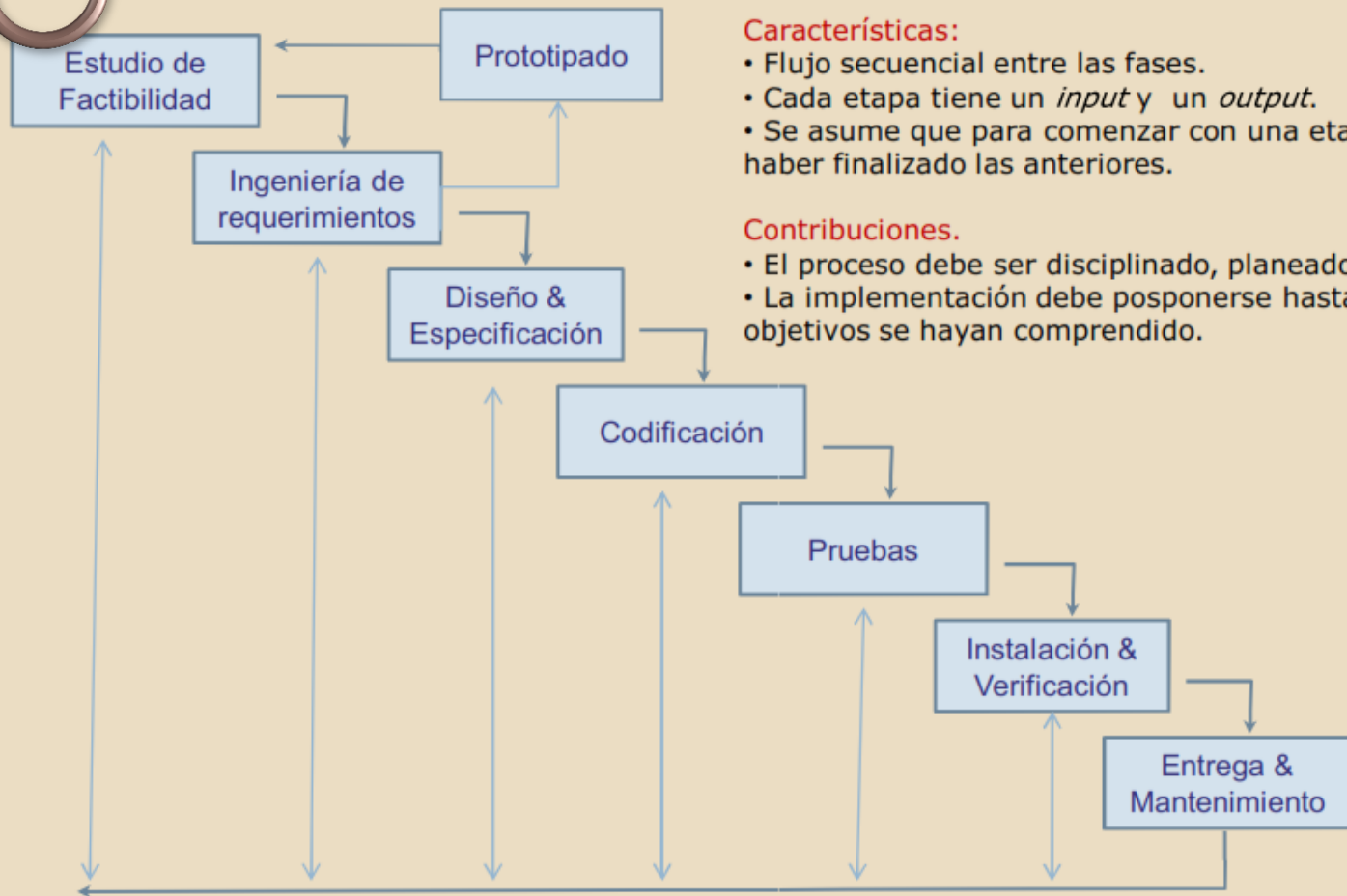




ADOO.

Estrategia 3: Tipos de Modelos de Ciclos de Vida.

LCM: Modelo de Ciclos de Vida en Cascada.



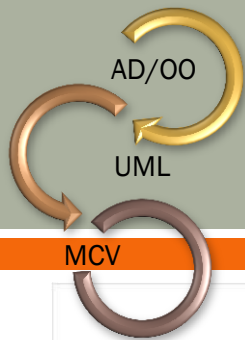
Características:

- Flujo secuencial entre las fases.
- Cada etapa tiene un *input* y un *output*.
- Se asume que para comenzar con una etapa deben haber finalizado las anteriores.

Contribuciones.

- El proceso debe ser disciplinado, planeado y gerenciado.
- La implementación debe posponerse hasta que los objetivos se hayan comprendido.

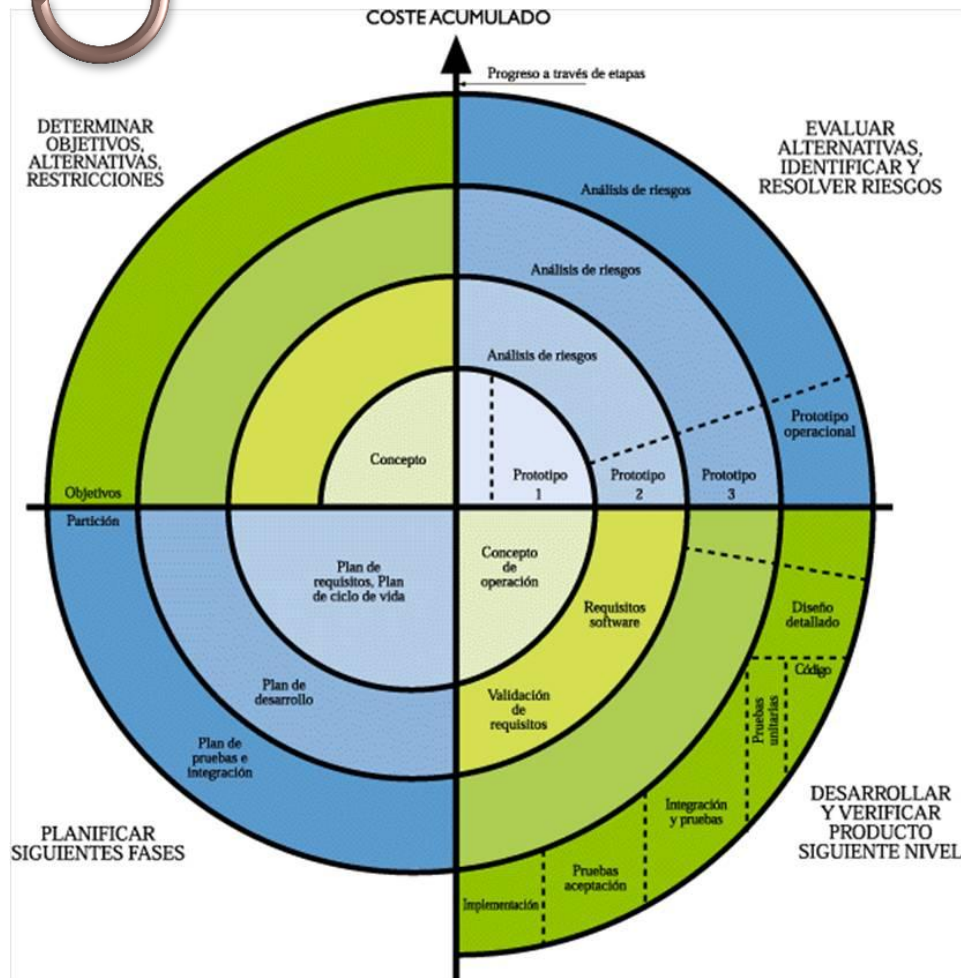




ADOO.

Estrategia 3: Tipos de Modelos de Ciclos de Vida.

LCM: Modelo en Espiral (Spiral).

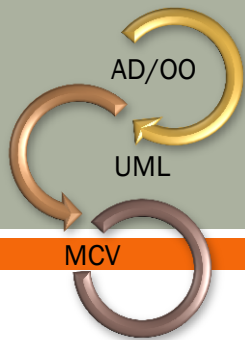


B. Boehm definió un modelo evolutivo para el desarrollo rápido de versiones incrementales denominado Modelo en Espiral. En este modelo se parte de un concepto básico, sobre el cual se desarrolla rápidamente un sencillo prototipo que permite confirmar que el concepto es válido.

A partir de aquí se va incrementando la funcionalidad y validando en cada paso que estamos en la línea correcta, evolucionando el sistema de forma consistente.

Este es un modelo especialmente adecuado cuando tratamos con proyectos innovadores o donde los requisitos no están muy claros

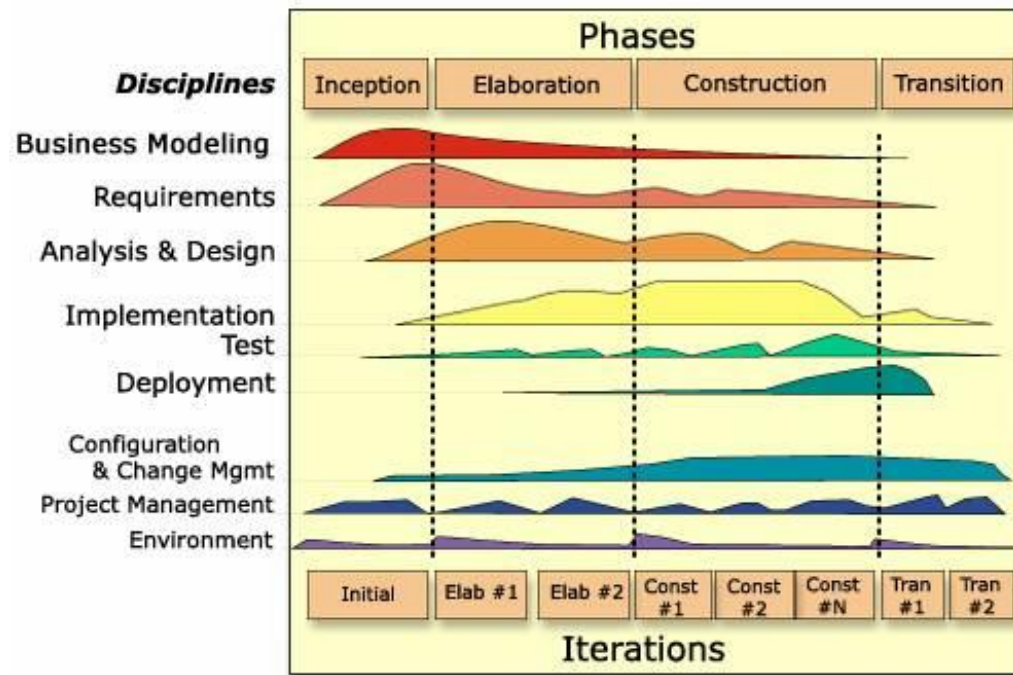




ADOO.

Estrategia 3: Tipos de Modelos de Ciclos de Vida.

LCM: EL Proceso Unificado de Desarrollo e Software.



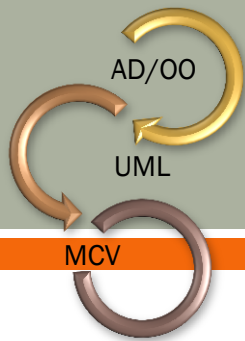
El Proceso Unificado de Desarrollo Software o simplemente **Proceso Unificado** es un marco de desarrollo de software que se caracteriza por estar dirigido por **casos de uso**, centrado en la **arquitectura** y por ser **iterativo e incremental**.

Lenguaje unificado de modelado

El Lenguaje unificado de modelado no es el sucesor de la oleada de métodos de análisis y diseño orientados a objetos que surgió a finales de la década de los 1980 y principios de la siguiente.

El **UML** unifica, sobre todo, los métodos de Booch, Rumbaugh, Brühl (OMT) y Jacobson, pero su alcance ha llegado a formar parte fundamental de la Ingeniería de Software tras su estandarización en 1997 con el **OMG** (Object Management Group o Grupo de Administración de Objetos).





ADOO.

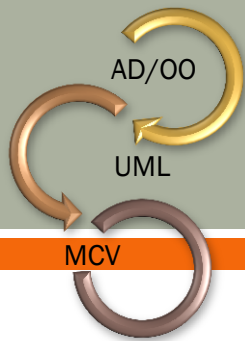
Conclusiones: Análisis & Diseño /OO.

¿Por que analizar y diseñar?

¿Por qué analizar y diseñar?

En resumidas cuentas, la cuestión fundamental del desarrollo del software es la escritura del código. Después de todo, los diagramas son solo imágenes agradables. Ningún usuario va a agradecer la belleza de los dibujos, lo que el usuario quiere es software que funcione (UML Gota a Gota, Addison_Wesley). Por lo tanto, cuando considere usar el UML es importante preguntarse por qué lo hará y como le ayudara a usted cuando llegue el momento de escribir el código. No existe una evidencia empírica adecuada que demuestre si estas técnicas son buenas o malas, pero lo que si es cierto es que es de considerable ayuda para las etapas de mantenimiento en proyectos de mediana/avanzada envergadura.





AD/OO. Bibliografía.

Referencia bibliográfica;

- **Analisis y Diseño Orientado a Objetos de Sistemas Usando UML.** Simon Bennett, Steve Mc. Robb y Ray Farmer. 3ra. Edicion. Editorial Mc. Graw Hill. Impreso en India. 2006. 620pp.
- **UML y Patrones.** Craig Larman. 2da. Edicion. Editorial Prentice Hall. Impreso en España. 2002. 590 pp.

Mg. Jorge Medianero Acosta
Docente FIIS / UNI

