



Universidad Nacional Autónoma de México
División del Sistema de Universidad Abierta [SUA - FCA]
Apuntes de la Materia: INFORMÁTICA V

Por: Prof. Ramon Castro Liceaga
Enero de 2006

Objetivo

Esta asignatura se imparte en el sexto semestre de la carrera de Licenciado en Informática y tiene el propósito de enseñar a los alumnos **a hacer buenos análisis y diseños en sistemas de información** para las organizaciones.

Introducción

En las nuevas versiones de Planes de estudio de la materia, se recomienda que este proceso se enfoque al paradigma de la programación orientada a objetos ya que este garantiza un nuevo estilo de ingeniería de software.

Apunte de cada unidad.

1.- EVOLUCION DEL DESARROLLO DE SISTEMAS

Temas:

- 1.1.- Conceptos y Evolución de la Teoría General de los Sistemas**
- 1.2.- Aplicación de los sistemas de Información**
- 1.3.- Concepto de Análisis y diseño**
- 1.4.- Ciclo de Vida de los sistemas**

Resumen:

En este tema destacamos los conceptos teóricos básicos que debes conocer sobre los sistemas de información en las organizaciones.

El origen de la teoría de los sistemas radica en la biología. El primer expositor de la esta teoría (padre de los sistemas) fue el biólogo Ludwing Von Bertalanffy en 1950 y en 1968 (año de revoluciones), e intentó lograr una metodología integradora sistémica (paradigma de los sistemas) para el tratamiento de los nuevos problemas científicos. Hoy en la actualidad esta teoría es un punto de apoyo muy importante para desarrollar nuevas ciencias como la informática cuántica, nanotecnología, teletransportación. realidad virtual, entre otras, y ha



aportado nuevos conceptos como : qbits (bits cuánticos), multiprocesamiento, reingeniería, análisis y desarrollo de sistemas, etc.

En el análisis, un investigador científico observa la problemática del mundo real, identifica metas globales, las perspectivas, las necesidades y requerimientos del cliente, la planificación presupuestal, lineamientos del mercado y el medio ambiente y otros puntos que puedan ayudar a la identificación y desarrollo de un proyecto o sistema de información.

El diseño de sistemas define un proceso de aplicación de ciertas técnicas y principios con el propósito de definir un proceso o sistema con suficientes detalles como para permitir su interpretación y realización física (operación). Deberás de considerar los pasos del “Ciclo de vida de sistemas” siguientes: Definición del problema, análisis del sistema, diseño del sistema, análisis de programación, preparación de programas, instalación y mantenimiento de programas

En este capítulo solo se hace mención de este ciclo de vida y en los próximos serás capaz de aplicarlos en un sistema de información que tu mismo desarrollarás.

2.- CARACTERISTICAS DE LOS PRODUCTOS DE SOFTWARE EN EL ANALISIS Y DE SARROLLO DE SISTEMAS

Temas:

2.1.- Herramientas para el análisis y desarrollo de software (CASE)

2.2.- prototipos, reportadores, etc.

2.3.- Factores externos:

2.3.1.- Correctez

2.3.2.- Robustez

2.3.3.- Extendibilidad

2.3.4.- Reusabilidad

2.3.5.- Compatibilidad

2.3.6.- Eficiencia

2.3.7.- Portabilidad

2.3.8.- Verifiabilidad

2.3.9.- Integridad

Resumen:

Este tema se complementa con el anterior ya que deberás conocer cuales son las herramientas mas importantes del desarrollo de software y cual es la calidad que deben tener los sistemas.



CASE es una combinación de herramientas de software y de metodología enfocadas básicamente a la productividad individual del profesional de desarrollo de software, su siglas son : Computer-Aided Software Engineering. Existen otras herramientas como los reportadores que son de gran utilidad para explotar la información de las Bases de Datos.

En la actualidad la calidad del software se considera como un objetivo estratégico de negocios, donde la calidad es una condición muy importante para lograr la eficiencia, para reforzar el trabajo y mejorar a la vez la productividad. Los atributos de calidad de un producto de software se dividen en internos y externos. Los internos se refieren a que los sistemas sean correctos, confiables, robustos, amigables, verificables y mantenibles. Los externos se refiere a que los sistemas sean reparables, reutilizable, portable, comprensible, deben tener interoperabilidad, productividad, oportunidad y con transparencia.

3.- ANALISIS DE SISTEMAS ORIENTADO A OBJETOS

Temas:

3.1.- Conceptos y metodologías para el análisis de sistemas

3.2.- Modelo conceptual de sistemas

3.3.- Interacción analista-usuario

3.4.- Identificación del problema

3.5.- Determinación de requerimientos

3.6.- Análisis costo-beneficio

3.7.- Estudio de Factibilidad

Resumen:

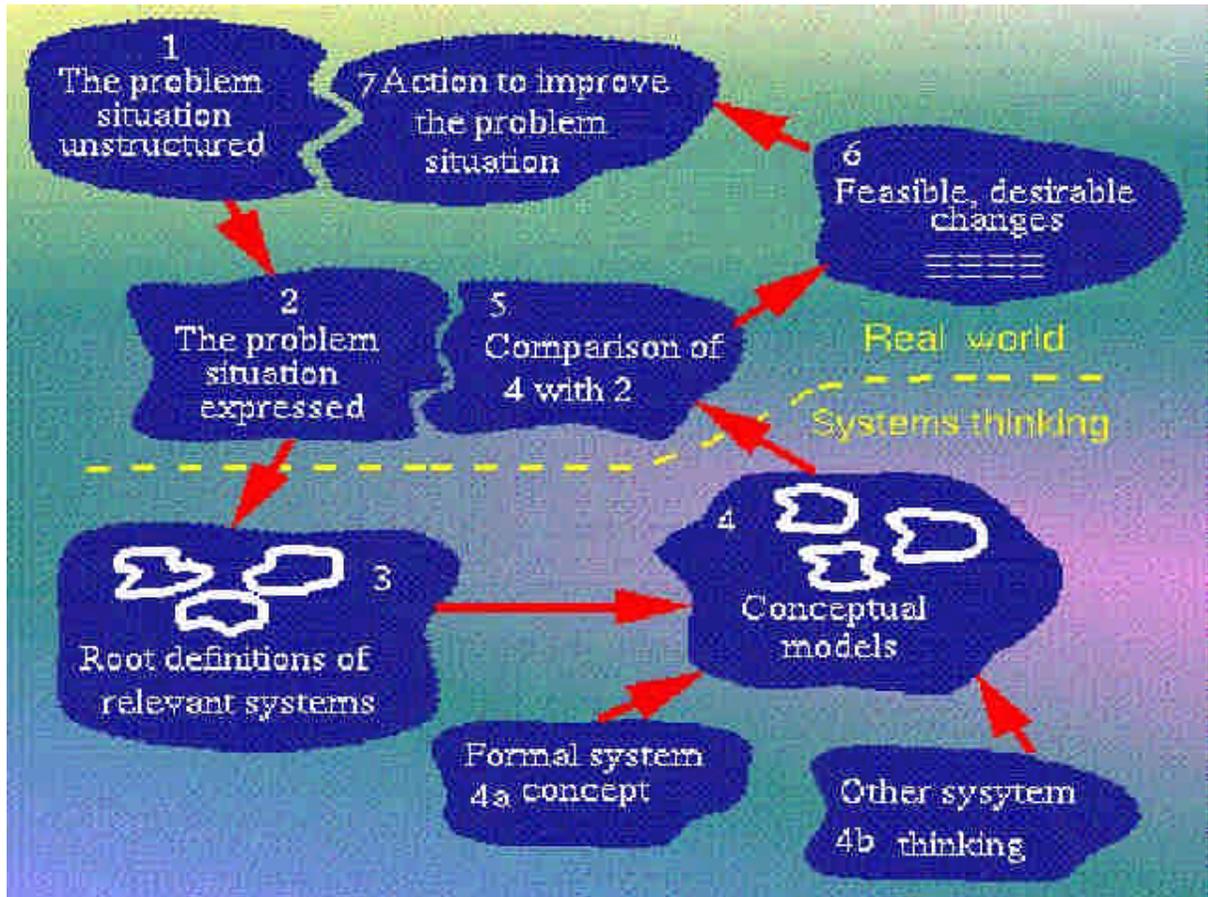
Una metodología es un conjunto de métodos empleados para el desarrollo de sistemas automatizados. Una metodología completa es algo más que una notación, un proceso, y herramientas. Además de una "notación, de un proceso, y de herramientas," estas "metodologías completas" proporcionan: Guías para estimar costos, manejo del proyecto en las tareas y entregas, medidas y métricas, formas definidas y dirección en las entregas de la construcción, políticas y procedimientos para garantizar la calidad del software, descripciones de los roles y programas de entrenamiento detallados, ejemplos totalmente trabajados, ejercicios de entrenamiento, técnicas para adaptar el método, y técnicas definidas. Las metodogías también han evolucionado. Inicialmente hubo un periodo de Desarrollo Convencional, después surge el Desarrollo Estructurado y en la actualidad aparece el paradigma de la Orientación a Objetos como un nuevo enfoque en la ingeniería de software.

A la fecha se han desarrollado muchísimas metodología enfocadas a la Orientación a Objetos, en esta materia nos enfocaremos a la propuesta por el profesor Peter Checkland (SSM) Soft Systems Methodology. Esta metodología se conoce también como "metodología soft" y parte del análisis de



una situación problemática no estructurada, el analista estructura el problema y lo refleja en un cuadro rico, realiza una definición esencial corta del problema, de la cual se deriva un modelo conceptual (sistema de actividades humanas), se hace una comparación entre lo conceptual y lo real, se realizan cambios y se toman las acciones para resolver el problema.

Ayuda didáctica: Metodología SSM



A partir del conocimiento de esta metodología ya podrás dar solución de problemas y tendrás los conceptos básicos para pasar al nivel del modelado.

4.- UNIFIED MODELING LANGUAGE (UML)

Temas:

- 4.1.- Que es UML
- 4.2.- El Paradigma Orientado a Objetos usando UML
- 4.3.- Fundamentos del Modelado OO
- 4.4.- Diagrama de Casos de Uso
- 4.5.- Diagramas de Interacción
- 4.6.- Diagrama de Clases



4.7.- Diagramas de Estados / Diagramas de Actividad

4.8.- Diagrama de Componentes

4.9.- Diagrama de Distribución

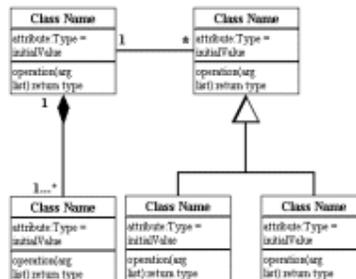
4.10.- Proceso de Desarrollo de SW basado en UML

Resumen:

El porqué de modelar. Un modelo es una simplificación de la realidad. Se construyen modelos de sistemas complejos porque no se puede comprender un sistema en su totalidad. El decidir qué modelo crear tiene una influencia profunda en cómo un problema es atacado y cómo una solución es formada. Cada modelo puede ser expresado a diferentes niveles de precisión. Los mejores modelos están conectados a la realidad. No hay modelo sencillo suficiente. Cada sistema no trivial está mejor enfocado mediante un conjunto pequeño de modelos cercanamente independientes.

El UML es un lenguaje estándar para escribir proyectos de software. El UML puede ser usado para visualizar, especificar, construir, y documentar los elementos de un sistema de software. Proporciona una manera estándar de escribir documentación del sistema (cosas conceptuales: procesos de negocio y funciones del sistema; cosas concretas: clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables). Las técnicas de modelado en UML son:

Diagrama de clases. Clases, interfaces, colaboraciones, relaciones. Describen la estructura estática de un sistema



Diagramas de Paquetes: Subclase de diagramas de clases.

Para cuando se desean técnicas separadas para relacionar elementos.

Diagrama elementos de un sistema relacionándolos en grupos separados, para minimizar dependencias entre paquetes.

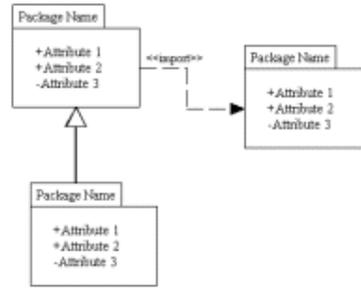


Diagrama de objetos. Objetos, relaciones. Describe la estructura estática en un momento en particular. Claves para ejecutar test de seguridad.

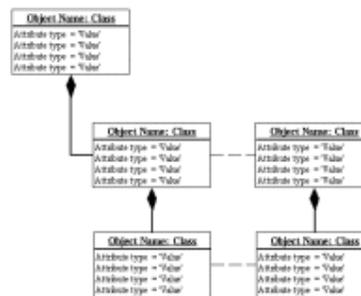
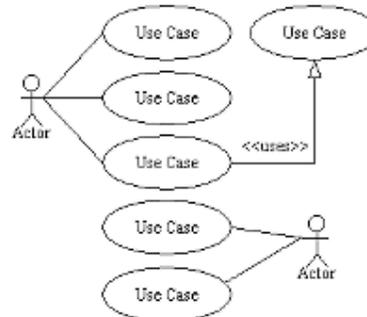


Diagrama de casos de uso. Organiza el comportamiento del sistema. Define la funcionalidad del sistema y sus actores.



Diagramas de interacción (secuencia y colaboración). Enfoque al intercambio de mensajes. Define interacciones entre clases en términos de intercambio de mensajes en el tiempo.

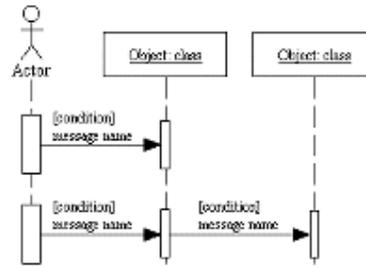


Diagrama de Diagramas de Colaboración: Representa interacciones entre objetos como una serie de secuencias de mensajes. Describe estructura y dinámica alrededor del sistema.

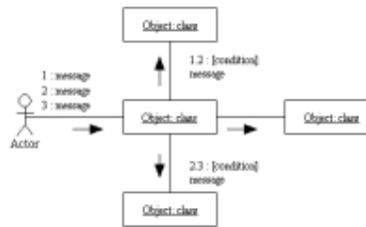


Diagrama de estados. Enfoque en los cambios del sistema dirigidos por eventos. Describe la conducta dinámica de un sistema en respuesta a estímulos externos. Usado para modelar objetos reactivos a eventos específicos.

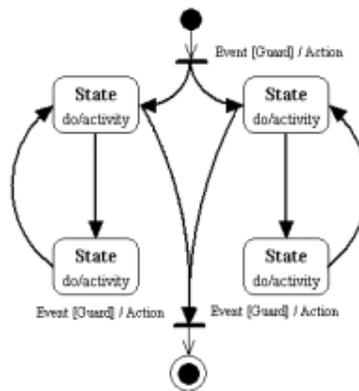


Diagrama de actividad. Flujo de control de actividad a actividad. Ilustra la naturaleza dinámica de un sistema, modelando el flujo de control entre



actividad y actividad. Típicamente usados para modelar flujos de negocios u operaciones internas.

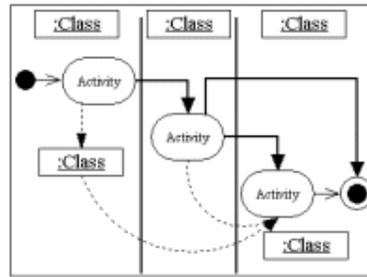


Diagrama de componentes. Componentes, relaciones. Describe la organización de los componentes físicos del software, incluyendo código fuente, código binario y ejecutables.

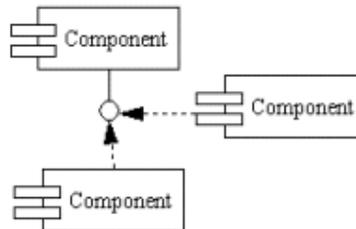
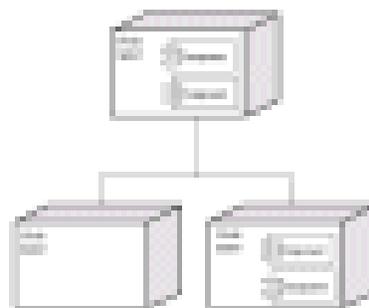


Diagrama de producción o despliegue. Nodos, relaciones. Describe los recursos físicos en un sistema, incluyendo nodos, componentes y conexiones.



Con estas herramientas ya podrás diseñar y desarrollar tu sistema.

5- DESARROLLO DE SISTEMAS

Temas:



5.1.- Modelo de datos

5.2.- Modelo de procedimientos con diagrama de flujo (Sistema actual vs. Nuevo Sistema)

5.4.- Diseño de Base de Datos

5.5.- Diagrama de Entidad Relación

5.6.- Diagrama de Transición de Estados

5.7.- Diccionario de Datos

5.8.- Diagrama de contexto

5.9.- Carta jerárquica del sistema

Resumen:

El desarrollo de sistemas consiste en desarrollar y documentar cada uno de los subsistemas que forman parte un Sistema Integrado. Las fases son: Desarrollo del Sistema y Prueba del Sistema (que veremos en el punto siguiente).

El Sistema a desarrollar se basa en los recursos requeridos en la etapa diseño de una Base de Datos (pueden ser múltiples y/o relacionales). Se debe contar con la aprobación del requerimiento. El desarrollo comienza cuando se tiene seguridad que el requerimiento está en marcha de compra / alquiler.

Después se entra a una etapa de capacitación de programadores y Operadores. La finalidad es capacitar a los programadores es para que puedan desarrollar los programas especificados y definidos con la nueva configuración. La finalidad de capacitar operadores es para que participen en la instalación y comprobación de los equipos y los operen en la prueba de los subsistemas. Se especifican las necesidades de programación, La finalidad de este paso es dar a los programadores las instrucciones para la elaboración de los programas del Sistema en desarrollo.

En el diagrama de entidad – relación se describirá el diseño de la Base de Datos o Bases de datos que se utilizarán en el sistema. En la carta jerárquica del sistema y Diagramas de Flujo del Proceso de Datos se identificarán: Cada uno de los programas a desarrollar. Entradas. Archivos. Salidas. Posteriormente se deberá desarrollar los programas del sistema. Se tendrán en cuenta las especificaciones de las necesidades de programación. Abarca las tareas de: diagramación, codificación y compilación.

Se debe imprimir cada uno de los programas fuente en salidas de impresora denominada "Listado Fuente" para la documentación integral del sistema.

En la materia realizarás este proceso utilizando un lenguaje de programación orientado a objetos como : java, c++ o c#. entregando la documentación correspondiente.

6.- PRUEBA Y EVALUACION DEL SISTEMA

Temas:

6.1.- Pruebas por modulo



- 6.2.- Pruebas al sistema**
- 6.3.- Pruebas en paralelo**
- 6.4.- Instalacion**
- 6.5.- Implantación**
- 6.6.- Manual de usuario**
- 6.7.- Capacitacion**
- 6.8.- Evaluación y seguimiento del sistema.**

Resumen:

El ultimo punto del ciclo de vida de sistemas se refiere a la prueba y evaluación del sistema que elaboraste.

Conjuntamente probaremos la operatoria de los programas del Sistema con los datos de prueba correspondientes.

Esta prueba es necesaria y fundamental, ya que una compilación sin errores no significa que el programa funcione correctamente. Normalmente aparecen errores no detectados. Los datos de prueba deben permitir verificar en forma completa un programa.

Aprenderás a elaborar un Manual de procedimientos. El objetivo de este paso es elaborar la Carpeta de Programa que corresponde a cada uno de los programas del subsistema. La correcta y completa documentación del programa facilitan su mantenimiento y las modificaciones en el subsistema. Realizaremos la prueba del Subsistema; el objetivo de este paso es verificar que el subsistema funcione y proporcione los resultados esperados. En las pruebas se deben prever los casos normales y otros que provoquen errores. Referente a la documentación del subsistema podemos decir que el objetivo de este paso es documentar en forma completa el subsistema a poner en marcha. Esta documentación conformará la carpeta del sistema. Documentar el sistema permite: Concentrar en un solo documento toda la información referida al Sistema. el control de los procesos, la evaluación del rendimiento del Sistema, y el mantenimiento (preventivo y correctivo) del Sistema.



Resumen Final : Mapa conceptual de la Materia Informática V.

