



DISEÑO DE SOFTWARE

Especialista en contenido:	LCDA. GRETTEY LEAL LCDO. OSWALDO BONITO	AUTORIZADO POR VICE RECTORADO ACADÉMICO (FIRMA Y SELLO)
Fecha de elaboración:	JULIO, 1994	
Elaborado por:	LCDA. GRETTEY LEAL LCDO. OSWALDO BONITO	

UNIDAD I	OBJETIVO TERMINAL DE UNIDAD
FUNDAMENTOS DEL SOFTWARE	AL FINALIZAR ESTA UNIDAD, EL ESTUDIANTE ESTARÁ EN CAPACIDAD DE REALIZAR PROCESOS DE DISEÑOS, DIFERENCIAR LOS CONCEPTOS DE FUNDAMENTOS DE DISEÑO, DEL DISEÑO MODULAR, DISEÑO DE DATOS, DISEÑO ARQUITECTÓNICO, DISEÑO PROCEDIMENTAL, ETC.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico - practica con ejemplos ilustrativos delos conceptos vistos en clases.	
OBJETIVOS ESPECÍFICOS	
- Describir la fase de desarrollo y diseño de software, procesos de diseños. - Describir los tipos de módulos, tipos de programas - Describir las características del diseño modular efectivo. - Estudiar las características básicas para graficar el diseño.	
CONTENIDOS	
- fase de desarrollo y diseño de software - proceso de diseño - fundamentos del diseño - diseño modular y efectivo - diseño de datos, arquitectónico y procedimental - programación estructurada - herramientas graficas del diseño. - Herramientas tabulares del diseño. documentación del diseño.	

UNIDAD II	OBJETIVO TERMINAL DE UNIDAD
DISEÑO ORIENTADO AL FLUJO DE DATOS	EL ESTUDIANTE ESTARÁ EN CAPACIDAD DE REALIZAR UN ANALISIS DE TRANSFORMACIÓN Y TRANSACCIÓN.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico - practica con ejemplos ilustrativos delos conceptos vistos en clases.	
OBJETIVOS ESPECÍFICOS	
- Realizar análisis de Transformación - Realizar análisis de Transacción - Optimizar el diseño.	
CONTENIDOS	
- Análisis de Transformación, de Transacción. - Heurística de diseño - Post-procesamiento en el diseño. - Optimización del diseño.	

UNIDAD III	OBJETIVO TERMINAL DE UNIDAD
DISEÑO ORIENTADO A LA ESTRUCTURA DE DATOS	AL FINALIZAR ESTA UNIDAD EL ESTUDIANTE ESTARÁ EN CAPACIDAD DE REALIZARE CONSTRUCCIONES LOGICAS DE PROGRAMAS Y SISTEMAS, DESARROLLAR DIAGRAMAS JACKSON Y WARNIER.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico-práctica con ejemplos ilustrados de los conceptos vistos en clases.	
OBJETIVOS ESPECÍFICOS	
- Desarrollar sistemas según Jackson y Warnier. - Utilizar métodos de diseño CLP. - Realizar construcciones lógicas se programa y sistemas.	
CONTENIDOS	
- Diseños y estructuras de datos. - Consideraciones sobre el proceso de diseño. - Desarrollo del sistema de Jackson. - Construcciones lógicas de programas y sistemas. - Diagrama de Warnier. - Método de diseño CLP	

UNIDAD IV	OBJETIVO TERMINAL DE UNIDAD
DISEÑO ORIENTADO AL OBJETO	AL FINALIZAR ESTA UNIDAD EL ESTUDIANTE ESTARA EN CAPACIDAD DE DIFERENCIAR Y APLICAR LOS CONCEPTOS DE DISEÑO ORIENTADO A OBJETO Y DE APLICAR LOS METODOS DEL DISEÑO ORIENTADO A OBJETO.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico-práctica con ejemplos ilustrados de los conceptos vistos en clases.	
OBJETIVOS ESPECÍFICOS	
- Conocer los conceptos de diseños orientados a objeto. - Aplicar los métodos de diseño orientado a objeto. - Definir problemas y estrategias.	
CONTENIDOS	
- Origen del diseño orientado a objeto. - Concepto del diseño orientado a objeto. - Métodos del diseño orientado a objeto. - Definición del problema. - Estrategia informal. - Formación de la estrategia. - Método alternativo.	

UNIDAD V	OBJETIVO TERMINAL DE UNIDAD
DISEÑO DEL TIEMPO REAL	AL FINALIZAR ESTA UNIDAD EL ESTUDIANTE ESTARÁ EN CAPACIDAD DE REALIZAR DISEÑO DE TIEMPO REAL; APLICAR LOS METODOS ORIENTADOS AL FLUJO DE DATOS.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico-práctica con ejemplos ilustrados de los conceptos vistos en clases.	
OBJETIVOS ESPECÍFICOS	
- Describir los métodos de diseño de software y orientados al flujo de datos. - Realizar diseño de tiempo real.	
CONTENIDOS	
- Consideraciones sobre los sistemas. - Sistema de tiempo real. - Análisis de sistema de tiempo real. - Métodos de diseño de software - Método de diseño orientado al flujo de datos.	

UNIDAD VI	OBJETIVO TERMINAL DE UNIDAD
TECNICA DE PRUEBA DE SOFTWARE	AL FINALIZAR ESTA UNIDAD, EL ESTUDIANTE ESTARA EN CAPACIDAD DE REALIZAR LOS DIFERENTES TIPOS DE PRUEBA AL SOFTWARE, APLICAR LAS ESTRATEGIAS DE PRUEBA DE SOFTWARE.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico-práctica con ejemplos ilustrados de los conceptos vistos en clases.	
OBJETIVOS ESPECÍFICOS	
- Realizar pruebas de software. - Conocer herramientas automáticas de prueba - Aplicar las estrategias de prueba de software	
CONTENIDOS	
- Fundamentos de la prueba de software. - Prueba de caja blanca, el camino básico, de bucles, de caja negra, de corrección. - Herramientas automáticas de prueba. - Enfoque estratégico de la prueba de software. - Prueba de unidad, de integración, de validación y del sistema.	

UNIDAD VII	OBJETIVO TERMINAL DE UNIDAD
MANTENIMIENTO DE SOFTWARE	AL FINALIZAR ESTA UNIDAD EL ESTUDIANTE ESTARÁ EN CAPACIDAD DE APLICAR EL MANTENIMIENTO AL SOFTWARE.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico-práctica con ejemplos ilustrados de los conceptos vistos en clases y asignación de trabajos.	
OBJETIVOS ESPECÍFICOS	
Realizar mantenimiento de software.	
CONTENIDOS	
Características, facilidad, tareas, efectos y aspectos del mantenimiento.	

UNIDAD VIII	OBJETIVO TERMINAL DE UNIDAD
COMPILADORES	AL FINALIZAR ESTA UNIDAD EL ESTUDIANTE ESTARÁ EN CAPACIDAD DE CONSTRUIR UN COMPILADOS.
DURACIÓN	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Clases Teórico-práctica y asignación de trabajo.	
OBJETIVOS ESPECÍFICOS	
- Realizar la fase de análisis sintáctico, léxico. - Incorporar tablas de símbolos. - Definir estrategias para la asignación de memoria. - Generar códigos intermedios.	
CONTENIDOS	
- Compiladores - Análisis de programas fuentes, fase de un compilador, agrupamiento de las fases, herramientas para la construcción de compiladores. Compilador sencillo de una sola pasada, ambiente para el momento de la ejecución, aspectos del lenguaje fuente, organización de la memoria, generación de código intermedio.	