



Curso «Análisis y diseño de sistemas»

Objetivo	
	Facilitar el proceso de gestión para la creación de un sistema o software previendo cambios en un futuro y mejorar la administración de éste.
Contenido temático	
	<ol style="list-style-type: none">1. Tecnología de objetos<ul style="list-style-type: none">▪ Diferencia entre análisis y diseño▪ Análisis y diseño orientado a objetos▪ Objetos y clases▪ Práctica inicial de análisis y diseño2. El ciclo de vida y el plan de trabajo con base en el proceso unificado<ul style="list-style-type: none">▪ El ciclo de vida▪ Fases e iteraciones▪ Artefactos y UML en el proceso unificado▪ Responsabilidades (trabajadores)▪ Disciplinas de ingeniería y de soporte3. La importancia del modelado visual4. Antecedentes de UML5. Modelo de casos de uso<ul style="list-style-type: none">▪ Actores▪ Casos de uso▪ Diagrama de casos de uso▪ Paquetes de casos de uso▪ Relaciones▪ Puntos de extensión▪ Paquetes de casos de uso6. Especificación de casos de uso<ul style="list-style-type: none">▪ Documentación de un caso de uso▪ Caso de uso de alto nivel▪ Flujos primarios, alternos y excepcionales▪ Precondiciones y poscondiciones▪ Requerimientos especiales del caso de uso▪ Escenarios▪ Las pruebas y los casos de uso7. Modelo conceptual<ul style="list-style-type: none">▪ Conceptos▪ Atributos▪ Relación de asociación▪ Diagrama del modelo conceptual▪ Identificación de conceptos mediante un análisis de casos de uso8. Diagramas de secuencia



- Clases y objetos
- Línea de vida
- Foco de control
- Mensajes y operaciones
- Diagrama de secuencia
- Diagrama de colaboración
- Diferencias entre el diagrama de colaboración y de secuencia
- Impacto del diagrama de interacción en el diagrama de clases

9. Patrones de asignación de responsabilidades

- Qué son los patrones
- Patrones para la asignación de responsabilidades
- Alta cohesión y bajo acoplamiento
- Diseño en tres capas

10. Diagramas de clases

- Clases
- Atributos
- Operaciones
- Alcance de atributos y operaciones
- Relaciones de asociación, agregación y dependencia
- Generalización: la implementación de la herencia
- Visibilidad entre clases
- Navegabilidad
- Multiplicidad
- Completando el diagrama de clases mediante el diagrama de interacción
- Paquetes de clases

11. Diagramas de componentes

- Componentes
- Interfaces
- La interfaz en el diagrama de clases
- La interfaz en el diagrama de componentes
- Relación de realización
- Tipos de componentes
- Dependencias

12. Diagramas de distribución

- Nodos
- Asociaciones entre nodos
- Dispositivos
- Diagrama de distribución

13. Implementación en el lenguaje seleccionado

- Interpretación del diagrama de clases
- Interpretación del diagrama de



	<p>secuencia</p> <ul style="list-style-type: none">▪ Interpretación del diagrama de componentes <p>14. Generación de código</p> <ul style="list-style-type: none">▪ Uso de herramientas CASE para la generación de código▪ Generación de código▪ Ingeniería inversa▪ Round Trip Engineering
Metodología	
	Exposición por parte del instructor y práctica en sala de cómputo.
Duración	
	25 horas