

- REF: MT-001
- NOMBRE: INTRODUCCIÓN AL ANÁLISIS ORIENTADO A OBJETOS
- 25 HORAS

Objetivos:

Conocer el origen, beneficios, términos y técnicas comúnmente utilizados en la Orientación a Objetos, la influencia de la Orientación a Objetos en los bloques principales de la construcción de software, así como su influencia sobre las aplicaciones nuevas y las ya existentes.

Dirigido:

A todos aquellos desarrolladores (o profesionales de las Tecnologías de la Información que no hayan programado previamente) que posteriormente deseen abordar el estudio de un Lenguaje Orientado a Objetos, como Smalltalk, C++, Java, Borland Delphi, Phyton, y otros.

Requisitos:

- *Conocimientos básicos de programación estructurada. No es necesario conocer previamente la Programación Orientada a Objetos, ya que se adquieren los fundamentos de la misma.*
- *Aunque no es imprescindible, es muy recomendable conocer algún lenguaje procedural (como Pascal, C, Basic, Cobol, PL1) y tener nociones básicas de programación (programa, dato, variable, flujo, código fuente, código ejecutable, implementación, pruebas).*

Contenido:

INTRODUCCIÓN AL ANÁLISIS ORIENTADO A OBJETOS

- 1 EL PROBLEMA DE LA CALIDAD DEL SOFTWARE Y EL ORIGEN DE LA POO**
- 2 FACTORES DE CALIDAD DEL SOFTWARE**
- 3 LOS CONCEPTOS BÁSICOS**
 - 3.1 Mecanismos básicos
 - 3.1.1 Objetos
 - 3.1.2 Comunicación entre objetos
 - 3.1.3 Mensajes y métodos
 - 3.1.4 Clases, subclasses y objetos
 - 3.1.5 Herencia
 - 3.1.6 Herencia simple
 - 3.1.7 Herencia múltiple
 - 3.2 Conceptos clave
 - 3.2.1 Encapsulación
 - 3.2.2 Abstracción
 - 3.2.3 Polimorfismo
 - 3.2.4 Persistencia
 - 3.3 Términos técnicos relacionados
 - 3.3.1 Ligadura dinámica
 - 3.3.2 Programación visual
 - 3.3.3 Bibliotecas de clases
 - 3.3.4 Sistemas controlados por eventos
 - 3.4 Correlación entre el modelo relacional y el OO

4 LOS PROBLEMAS DE LA POO

- 4.1 ¿Por qué utilizar la Programación Orientada a Objetos?
- 4.2 Problemas de la POO

5 LAS VENTAJAS DE LA POO

- 5.1 Gestión de la complejidad
- 5.2 Flexibilidad en el desarrollo del software
- 5.3 Reutilización
- 5.4 Aumento de la productividad
- 5.5 Extensibilidad y mantenibilidad
- 5.6 Programación por el usuario
- 5.7 Lenguajes orientados a objeto
- 5.8 Bases de datos orientadas a objeto
- 5.9 Interfaces de usuario

6 ASPECTOS AVANZADOS

- 6.1 Sobrecarga de métodos. Sobreescribir métodos.
- 6.2 Ligadura dinámica frente a estática
- 6.3 Clases como objetos
- 6.4 Concurrencia
- 6.5** Estándares o normas

7 TEMAS PARA SABER MAS

- 7.1 Análisis y diseño orientado a objetos
- 7.2 Programación y mantenimiento
- 7.3 Herramientas
- 7.4 Ejemplos de aplicaciones orientadas a objetos
- 7.5** Tendencias futuras
- 7.6 Resumen
- 7.7 Ejercicios