



**LA FORMACIÓN ES LA CLAVE  
DEL ÉXITO**

# Guía del Curso

## Curso de Programación Orientado a Componentes

---

Modalidad de realización del curso: [Online](#)

Titulación: [Diploma acreditativo con las horas del curso](#)

---

### OBJETIVOS

Este Curso de Programación Orientado a Componentes le ofrece una formación especializada en la materia. Debemos saber que en la actualidad, en el mundo de la informática y las comunicaciones, es muy importante conocer la programación de sistemas informáticos, dentro del área profesional de sistemas y telemática. Por ello, con el presente curso se trata de aportar los conocimientos necesarios para conocer el diseño de elementos software con tecnologías basadas en componentes y la implementación e integración de elementos software con tecnologías basadas en componentes.

### CONTENIDOS

#### UNIDAD FORMATIVA 1. DISEÑO DE ELEMENTOS SOFTWARE CON TECNOLOGÍAS BASADAS EN COMPONENTES

#### UNIDAD DIDÁCTICA 1. LA ORIENTACIÓN A OBJETOS

1. Principios de la orientación a objetos. Comparación con la programación estructurada:
2. - Ocultación de información (information hiding)
3. - El tipo abstracto de datos (ADT). Encapsulado de datos
4. - Paso de mensajes

5. Conceptos básicos de orientación a objetos:
6. - Clases:
7. - Atributos, variables de estado y variables de clase
8. - Métodos. Requisitos e invariantes
9. - Gestión de excepciones
10. - Agregación de clases
11. - Objetos:
12. - Creación y destrucción de objetos
13. - Llamada a métodos de un objeto
14. - Visibilidad y uso de las variables de estado
15. - Referencias a objetos
16. - Persistencia de objetos
17. - Optimización de memoria y recolección de basura (garbage collection)
18. - Herencia:
19. - Concepto de herencia. Superclases y subclases
20. - Herencia múltiple
21. - Clases abstractas
22. - Tipos de herencia: herencia de implementación, herencia de interfaces y de tipos y otros tipos de herencia
23. - Polimorfismo y enlace dinámico (dynamic binding)
24. - Directrices para el uso correcto de la herencia
25. - Modularidad:
26. - Librerías de clases. Ámbito de utilización de nombres
27. - Ventajas de la utilización de módulos o paquetes
28. - Genericidad y sobrecarga:
29. - Concepto de genericidad
30. - Concepto de Sobrecarga. Tipos de sobrecarga
31. - Comparación entre genericidad y sobrecarga
32. Desarrollo orientado a objetos:
33. - Lenguajes de desarrollo orientado a objetos de uso común
34. - Herramientas de desarrollo
35. Lenguajes de modelización en el desarrollo orientado a objetos:
36. - El lenguaje unificado de modelado (UML)

37. - Diagramas para la modelización de sistemas orientados a objetos

## UNIDAD DIDÁCTICA 2. LA ORIENTACIÓN A COMPONENTES

1. Fundamentos conceptuales:
2. - Definición de componente
3. - Comparación entre componentes y objetos
4. - Módulos
5. - Interfaces:
6. - Tipos de interfaces
7. - Versionado de interfaces
8. - Interfaces como contratos
9. - Escalado de componentes
10. - Estado de componentes
11. Arquitecturas de componentes:
12. - Basadas en objetos. Composición y uso de objetos
13. - Multicapa
14. - Basadas en middleware
15. - Basadas en objetos distribuidos
16. Diseño de componentes:
17. - Principios de diseño de componentes:
18. - Dependencias no cíclicas
19. - Principio "open/closed".
20. - Reusabilidad
21. - Configurabilidad
22. - Abstracción
23. - Dependencias
24. - Técnicas de reusabilidad:
25. - Patrones
26. - Librerías
27. - Interfaces
28. - Protocolos y esquemas de mensajes
29. - Uso de lenguajes de programación
30. - Estructuras y jerarquías de estructuras

31. - Arquitecturas de sistemas
32. - Modelo de componente:
33. - Especificación de servicios: transacciones, seguridad, persistencia y acceso remoto
34. - Especificación de Interface
35. - Especificación de la implementación
36. - Especificación de las unidades de despliegue (modulos)
37. - Modelos de integración de componentes:
38. - Referencias e identidad de objetos, componentes e interfaces
39. - Servicios de localización
40. - Modelos de intercambio: objetos distribuidos, capa intermedia (Middleware) e interacción e integración mediante servicios web
41. - Comparación entre métodos de intercambio en las principales infraestructuras de componentes: OMG: CORBA, OMA, Java: JavaBeans, EJBs y Microsoft: COM, OLE/ActiveX, . NET
42. - Diagramación y documentación de componentes:
43. - Modelo de información: diagramas conceptuales, diagramas de arquitectura de componentes y diagramas de despliegue
44. - Modelo dinámico: diagramas de interacción y de actividad, diagramas de casos de uso y diagramas de estado

## UNIDAD FORMATIVA 2. IMPLEMENTACIÓN E INTEGRACIÓN DE ELEMENTOS SOFTWARE CON TECNOLOGÍAS BASADAS EN COMPONENTES

### UNIDAD DIDÁCTICA 1. DESARROLLO DE COMPONENTES

1. Lenguajes de desarrollo de componentes. :
2. - Comparativa con lenguajes orientados a objetos
3. - Lenguajes orientados a componentes:
4. - Descripción de interfaces
5. - Ensamblado
6. - Descripción de arquitectura
7. Requisitos principales del desarrollo orientado a componentes:

8. - Modularidad
9. - Despliegue independiente
10. - Reemplazabilidad
11. - Seguridad
12. - Separación entre interfaz e implementación
13. Infraestructuras (frameworks) de componentes:
14. - Modelos de infraestructuras de componentes:
15. - Orientados a conexión
16. - Orientados a contexto
17. - Orientados a aspectos
18. - Descripción de las infraestructuras de componentes de uso común:
19. - OMG: CORBA, OMA
20. - Java: JavaBeans, EJBs
21. - Microsoft: COM, OLE/ActiveX, .NET
22. Métodos de desarrollo de componentes:
23. - Uso de lenguajes orientados a objetos
24. - Selección de infraestructuras de componentes
25. Construcción de software mediante componentes:
26. - Definición de interfaces. Lenguajes de descripción de interfaces
27. - Reutilización de componentes
28. - Técnicas de ensamblado en infraestructuras de uso común
29. Técnicas específicas de desarrollo:
30. - Componentes en la capa de servidor web. Páginas dinámicas
31. - Componentes en la capa de servidor de aplicaciones
32. - Componentes en la capa de aplicación cliente:
33. - Componentes de interfaz gráfico
34. - Componentes orientados a documento
35. - Componentes en la capa de servicios web
36. - Componentes para dispositivos móviles
37. Herramientas para el desarrollo de componentes:
38. - Entornos integrados de desarrollo de componentes
39. - Configuración e instalación de herramientas de uso común:
40. - Entorno Java
41. - Entorno .NET

42. - Gestión del ciclo de vida en el desarrollo de componentes mediante herramientas de uso común:
43. - Uso de repositorios de componentes. Registro de componentes
44. - Reutilización de componentes para la construcción de sistemas software
45. - Definición de metadatos de componente. Descriptores de interfaces
46. - Modelo de seguridad
47. - Instalación de componentes
48. - Depuración y prueba de componentes

## UNIDAD DIDÁCTICA 2. COMPONENTES DISTRIBUIDOS

1. Programación distribuida en infraestructuras de uso común:
2. - Programación multihilo (multithreading)
3. - Comunicaciones síncronas y asíncronas
4. Modelos de intercambio:
5. - Llamadas a procedimientos remotos
6. - Orientados a mensajes
7. - Orientados a recursos



C/ San Lorenzo 2 - 2  
29001 Málaga



Tlf: 952 215 476  
Fax: 951 987 941



[www.academiaintegral.com.es](http://www.academiaintegral.com.es)  
E-mail: [info@academiaintegral.com.es](mailto:info@academiaintegral.com.es)