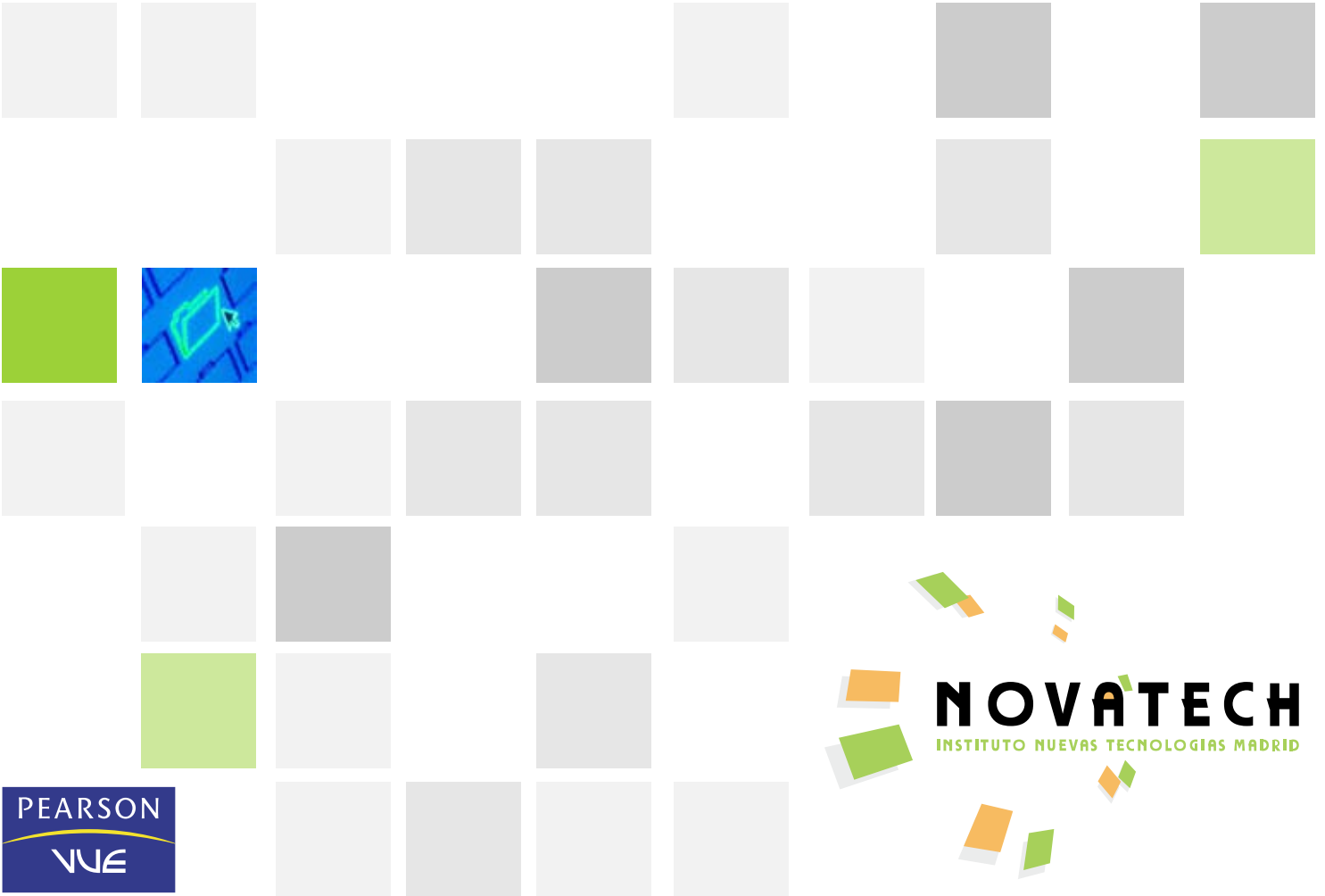


# Cursos Superiores

CURSO SUPERIOR EN PROGRAMACION UML, POO Y PATRONES

A photograph of a hand holding a computer mouse, positioned on the right side of the banner.

# Indice

## 03

### » Introducción

Dirigido a  
Prerrequisitos  
Modalidades de Impartición  
Formas de Pago  
Duración y Precios

## 05

### » Material

## 06

### » Aulas e Instalaciones

## 07

### » Temario

Curso Superior Programación  
UML, POO y Patrones



## 09

- » Beneficios para la empresa al disponer de personas certificadas
- » Formación para lograr una titulación con reconocimiento mundial

## 10

- » Comité Académico

## 11

- » Entidades Colaboradoras

## Introducción

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial.

La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación. Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades, métodos y otras cosas que veremos rápidamente para aclarar conceptos y dar una pequeña base que permita soltarnos un poco con este tipo de programación.

Los Patrones de Diseño (Design Patterns) son el esqueleto de soluciones a problemas comunes en el desarrollo de software.

## Dirigido a

Todas aquellas personas que tienen conocimientos básicos de programación y quieren especializarse en el lenguaje UML, y requieren que su formación esté orientada a POO (Programación Orientada a Objetos), Java, C++.

## Prerrequisitos

- Con la finalidad de que el alumno obtenga el máximo aprovechamiento de este curso de UML, es recomendable que el alumno cuente con los conceptos fundamentales de la programación orientada a objetos.
- También es bastante deseable que el profesional ostente experiencia previa en el uso de alguno de los lenguajes de programación orientado a objetos moderno.

## Formas de pago

**PAGO ÚNICO:** Un único pago del total del curso, con un 8 % de descuento sobre el precio final. Por transferencia o domiciliación bancaria.

**PAGO FRACCIONADO:** Pago fraccionado mediante domiciliación bancaria. Fraccionamiento hasta en tres pagos. Interés 0%.

**PAGO FINANCIADO:** Pago en 6 meses con interés 0%. Tarjeta Punto Oro. Sujeta a concesión.

**PAGO FINANCIADO:** A 12 meses. Consúltanos. Hasta un máximo de 3000 Euros.

*La formación en España está exenta de IVA*

## Modalidades de impartición

- ▶ **Presencial**
- ▶ **Semi-Presencial** (Blended)
- ▶ **E-learning.** Plataforma es.skillport. Acceso ilimitado durante un año. Consúltanos.

## Duración y Precios

Duración:	120 Horas
Precio:	1.490 Euros
Matrícula:	60 Euros*

\*no incluida en el precio del curso

## Material

Documentación incluida en Castellano, 3 libros:

» Para la parte de UML, “El Lenguaje Unificado de Modelado” redactada por los creadores del lenguaje de programación UML:

**Autores: Grady Booch, James Rumbaugh, Ivar Jacobson, compuesto por 3 tomos:**

- El Lenguaje Unificado de Modelado
- El Lenguaje Unificado de Modelado, Manual de Referencia
- El proceso unificado de desarrollo de Software

Incluye documentación en formato CDROM con vídeos, esquemas, diagramas etc elaborados por los propios autores / creadores del lenguaje UML.

» Para la parte de POO, “Introducción a la Programación con Orientación a Objetos”

**Autores: Camelia Muñoz Caro, Alfonso Niño Ramos, Aurora Vizcaíno Barceló.**

Este libro presenta una amplia introducción a la programación entroncada en el paradigma de la orientación a objetos, sin centrarse en el componente sintáctico de ningún lenguaje determinado.

Contiene numerosos ejemplos y ejercicios implementados en Java formalmente desarrollados y resueltos, susceptibles de ser utilizados en sesiones prácticas de laboratorio.

» Para la parte de Patrones de Diseño, “Patrones de Diseño”

**Autor: Gamma, E. Editorial: Addison Wesley.**



Instituto Novatech / Tel. 913 952 875 / [www.institutonovatech.com](http://www.institutonovatech.com)

## Aulas e Instalaciones

INSTITUTO NOVATECH pone a disposición de los asistentes a sus cursos instalaciones que incluyen aulas informatizadas con conexión a Internet por ADSL a 4Mb, máquinas de vending, y sala de exámenes y certificación.

### Oficinas Centrales:

#### Moncloa – Argüelles.

C/ Gaztambide, 89 (San Francisco de Sales, 12)  
Tel. 91.395.28.75

### Comunicaciones

- Metros: Islas Filipinas, Intercambiador de Moncloa.
- Autobuses: líneas C, 1, 12, 44, 888, N18.
- Zonas de parking: Hilarión Eslava, 51. General Ampudia, 8. San Francisco de Sales, 16. Zonas azules alrededor.

### Instalaciones

- Local con acceso desde calle particular.
- Sistema de Climatización Frío / Calor centralizado.
- Aislamiento frente al ruido.
- Atención personalizada permanente.
- Área de descanso con máquinas de vending, café y bebidas frías.

### Equipamiento en aula

- Equipos Portátiles Dell con Procesadores INTEL DUAL CORE T2300, 1GB RAM para impartición de cursos basados en Redes y Comunicaciones.
- Equipos última generación AMD 2,4 Ghz con 512 MB de RAM para impartición de cursos basados en Programación, Diseño y Desarrollo Web.
- Monitores TFT LG 17"



- Instalación en Red Corporativa de los equipos informáticos.
- Conexión a Internet con ADSL (2 / 4 Mbps).
- Sillas ergonómicas y regulables.
- Pizarras blancas Vileda.
- Vídeo proyector.
- 1 ordenador por alumno.
- 1 ordenador para el profesor.

Instituto Novatech / Tel. 913 952 875 / [www.institutonovatech.com](http://www.institutonovatech.com)

## Temario Curso Superior Programación UML, POO y Patrones

### 1) PROGRAMACIÓN EN UML (40 horas)

#### 1. Antecedentes de UML

#### 2. La Importancia del Modelado Visual

#### 3. Tecnología de Objetos

#### 4. El ciclo de vida y el plan de trabajo con base en RUP

- 4.1. El Ciclo de Vida
- 4.2. Fases e Iteraciones
- 4.3. Artefactos y UML en RUP
- 4.4. Responsabilidades
- 4.5. Disciplinas

#### 5. Modelo de Casos de Uso

- 5.1. Actores
- 5.2. Casos de Uso
- 5.3. Diagrama de Casos de Uso
- 5.4. Paquetes de Casos de Uso

#### 6. Flujos de Eventos

- 6.1. Documentación de un Caso de Uso
- 6.2. Flujos Primarios y Alternos
- 6.3. Análisis de Casos de Uso

#### 7. Modelo Conceptual

- 7.1. Conceptos
- 7.2. Atributos
- 7.3. Relación de Asociación
- 7.4. Diagrama del Modelo Conceptual

#### 8. Diagramas de Clases

- 8.1. Clases
- 8.2. Operaciones
- 8.3. Alcance de Atributos y Operaciones
- 8.4. Relaciones de Herencia, Agregación y Dependencia
- 8.5. Visibilidad entre Clases
- 8.6. Navegabilidad
- 8.7. Multiplicidad

#### 9. Diagramas de Secuencia

- 9.1. Clases y Objetos
- 9.2. Línea de Vida
- 9.3. Foco de Control
- 9.4. Mensajes y Operaciones
- 9.5. Diagrama de Secuencia
- 9.6. Diagrama de Colaboración
- 9.7. Diferencias entre el Diagrama de Colaboración y de Secuencia

#### 10. Diagramas de Componentes

- 10.1. Paquetes de Clases
- 10.2. Componentes
- 10.3. Interfaces
- 10.4. Tipos de Componentes
- 10.5. Dependencias

#### 11. Diagramas de Distribución

- 11.1. Nodos
- 11.2. Asociaciones entre Nodos
- 11.3. Diagrama de Distribución

#### 12. Implementación en Visual Basic (también puede realizarse en Java, C++ o Delphi)

- 12.1. Interpretación del Diagrama de Clases
- 12.2. Interpretación del Diagrama de Secuencia
- 12.3. Interpretación del Diagrama de Componentes

#### 13. Generación de Código en la Herramienta Seleccionada (Rational Rose, MS Visio, Select, Model Maker, etc.)

- 13.1. Uso de herramientas CASE para la Generación de Código
- 13.2. Generación de Código
- 13.3. Ingeniería Inversa
- 13.4. Round Trip Engineering



## Temario Curso Superior Programación UML, POO y Patrones

### 14. Patrones de Diseño

- 14.1. Patrones para la Asignación de Responsabilidades

### 15. Ejemplo y Caso Práctico

- 15.1. Desarrollo de un segundo Caso Práctico para Repasar los Conceptos Aprendidos: Este caso práctico puede ser desarrollado a partir de los requerimientos de un sistema del mismo cliente.

### 16. Diagramas de Estado

- 16.1. Estado
- 16.2. Transiciones
- 16.3. Tipos de Estados

### 17. Diagramas de Actividad

- 17.1. Actividades
- 17.2. Estados
- 17.3. Elementos del Diagrama de Actividad

## 2) PROGRAMACION ORIENTADA A OBJETOS (40 horas)

### UNIDAD DIDÁCTICA 1. FUNDAMENTOS.

#### 1. Introducción

#### 2. Programas sencillos

- 2.1 Fundamentos de la programación orientada a objetos
- 2.2 Introducción al tema de salida
- 2.3 La instrucción de asignación
- 2.4 Usando paquetes, clases y objetos

#### 3. Estructuración

- 3.1 Propiedades de un buen programa
- 3.2 Tipos
- 3.3 Repetición con bucles for
- 3.4 Construcción de métodos
- 3.5 Paso de parámetros
- 3.6 Pautas para el diseño de clases

#### 4. Cambios de estado

- 4.1 Entrada interactiva
- 4.2 Entrada y salida de archivos

- 4.3 Selección con if-then-else
- 4.4 Introducción a las excepciones
- 4.5 Creación y acceso a paquetes

### 5. Control de flujo

- 5.1 Bucles condicionales con while y do
- 5.2 La instrucción switch
- 5.3 Bucles condicionales y entrada de datos

### 6. Tablas unidimensionales y multidimensionales

- 6.1 Tablas unidimensionales
- 6.2 Tablas multidimensionales
- 6.3 Ordenación y búsqueda
- 6.4 Diccionarios

### 7. Formateo

- 7.1 Cadenas de caracteres y su manejo
- 7.2 Troceadores y la clase Text
- 7.3 Dar formato a números con el paquete de java.txt
- 7.4 Fechas, calendarios y tiempo

### UNIDAD DIDÁCTICA 2. POTENCIA.

#### 1. Objetos en marcha

- 1.1 La protección de los objetos
- 1.2 Diseño de un programa orientado a objetos
- 1.3 Propiedades de los objetos
- 1.4 Listas de objetos
- 1.5 Clases internas

#### 2. Abstracción y herencia

- 2.1 Dar potencia a las clases
- 2.2 Abstracción mediante interfaces
- 2.3 Herencia
- 2.4 Métodos y clases abstractos

#### 3. Interfaces gráficas de usuario

- 3.1 Introducción al awt
- 3.2 Posicionamiento de gráficos en una ventana
- 3.3 Posicionamiento de botones y otros componentes
- 3.4 Paneles y lienzos para el agrupamiento de componentes

## Temario Curso Superior Programación UML, POO y Patrones

### 4. Programación dirigida por sucesos

- 4.1 Interacción con campos de texto
- 4.2 Sucesos, oyentes y manipuladores
- 4.3 Gestión de sucesiones de sucesos

### 5. Applets

- 5.1 De aplicaciones a applets
- 5.2 Applets en visores
- 5.3 Imágenes y sonido

## 3) PATRONES DE DISEÑO (40 horas)

### 1. Patrones. Su historia, motivación y evolución

- 1.1 Qué es un patrón?
- 1.2 Identificando patrones
- 1.3 Representando patrones: las distintas partes que constituyen la descripción de un patrón
- 1.4 Conceptos de refactoring, antipatrones
- 1.5 Primeros ejemplos de patrones

### 2. Catálogo de Patrones listas de patrones ya armadas como:

- 2.1 GoF (Patrones de Diseño de la "Gang of Four")
- 2.2 Patrones J2EE (lista preparada por Sun)
- 2.3 TheServerSide.com (lista preparada por la comunidad de ese sitio)

### 3. Algunas Clasificaciones

- 3.1 Estructurales, Creacionales, de Conducta.
- 3.2 Clasificaciones por dominio de aplicación.
- 3.3 Después de una presentación general del tema, nos concentraremos en su aplicación, en particular en Java y J2EE. Descubriremos los patrones subyacentes en las ideas implementadas en JSP, y EJB, y otras tecnologías J2EE, y cómo aprovecharlas en nuestros sistemas.

### 4. Primeros ejemplos de Patrones

- 4.1 Singleton, Facade, Abstract Factory, Decorator, Factory Method, Builder

### 5. Patrones en la Persistencia

- 5.1 Concepto de Persistencia.
- 5.2 Patrón Data Access Object
- 5.3 Patrón Value Object
- 5.4 Patrón Service Locator
- 5.5 Implementaciones en Java, EJB, otros frameworks de persistencia

### 6. Patrones en la Capa de Presentación

- 6.1 Patrón Front Controller
- 6.2 Patrón View Helper
- 6.3 Patrón Intercepting Filter
- 6.4 Patrón Service-to-Worker
- 6.5 Patrón Composite View
- 6.6 Patrón Command
- 6.7 Patrón Abstract Factory
- 6.8 El Model View Controller
- 6.9 Implementaciones en JSP

### 7. Patrones en Aplicaciones Distribuidas

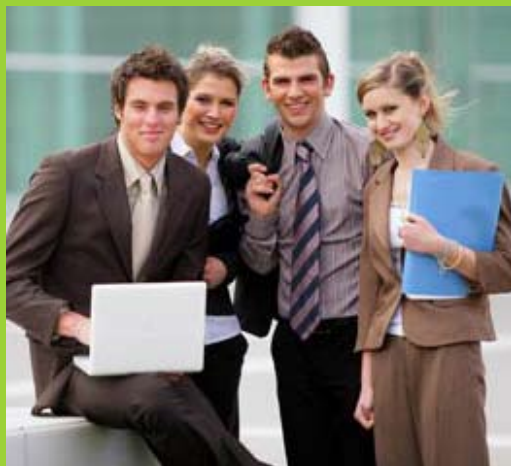
- 7.1 Los problemas a resolver en aplicaciones distribuidas
- 7.2 Escalabilidad
- 7.3 Llamadas sincrónicas y asincrónicas
- 7.4 Pasaje de Datos, Patrón Session Facade, Patrón Message Facade, Patrón Business Delegate
- 7.5 Aplicaciones del Value Object, Service Locator

### 8. Patrones en Seguridad

- 8.1 Patrón Single Access Point
- 8.2 Patrón Check Point
- 8.3 Patrón Role
- 8.4 Su implementación en JSP y EJB

### 9. Patrones en Integración

- 9.1 El problema de integrar distintas tecnologías y sistemas
- 9.2 Enterprise Application Integration
- 9.3 Patrón Integration Broker, Patrón Wrapper, Patrón Virtual Component, Patrón Data Mapping
- 9.4 Patrón Integration Mediator (de un paso, de múltiples pasos)
- 9.5 Aplicaciones del Data Access Object
- 9.6 Patrón Process Automator



### Beneficios para la empresa al disponer de personas certificadas:

- » Ahorro en tiempo y dinero.
- » Contar con personal calificado para evitar retrasos innecesarios en el funcionamiento de los sistemas de su empresa.
- » Eficacia en el desempeño de su trabajo y "satisfacción al cliente".
- » Eliminar demoras.
- » Facilidad en el reclutamiento de personal capacitado.

### Formación para lograr una titulación con reconocimiento mundial

INSTITUTO NOVATECH mantiene acuerdos de colaboración con prestigiosas Instituciones, Fundaciones y Organismos con el claro objetivo de participar activamente en el Progreso de la Sociedad de la Información.

INSTITUTO NOVATECH es además Socio Colaborador de los principales Proyectos Tecnológicos Mundiales, Cisco CNAP, Microsoft IT Academy, CompTIA, Google Adwords, Microsoft MOS, Oracle Partner Network, Linux LPI, etc.)

INSTITUTO NOVATECH es miembro de: Instituto de la Ingeniería de España (IIES), miembro de la Asociación Nacional de empresas de Electrónica, Tecnología de la Información y Telecomunicaciones de España (AETIC), miembro de la Asociación Internacional Comptia, socio colaborador del programa para la impartición de seminarios de Google AdWords, además de Colaborador de PUE, Proyecto Universidad Empresa.

INSTITUTO NOVATECH también es Centro Certificador Oficial PEARSON VUE, por lo que todos los asistentes a nuestros Programas Formativos pueden certificarse en nuestras instalaciones.

INSTITUTO NOVATECH es una marca registrada y patentada. Temario registrado y patentado por Instituto Novatech. Quedan reservados todos los derechos.



Instituto Novatech / Tel. 913 952 875 / [www.institutonovatech.com](http://www.institutonovatech.com)

## Comité Académico

INSTITUTO NOVATECH ha constituido un Comité Académico, de carácter asesor y consultivo, formado por las siguientes personalidades:

### D. SATURNINO DE LA PLAZA

- › Catedrático de Universidad
- › Ex-Rector de la Universidad Politécnica de Madrid
- › Ex-presidente de la Conferencia de Rectores de las Universidades Españolas (CRUE)

### D. OCTAVIO UÑA JUÁREZ

- › Catedrático de Sociología de la Universidad Rey Juan Carlos
- › Profesor de la Escuela Diplomática del Ministerio de Asuntos Exteriores
- › Catedrático de Teoría de la Comunicación de la Universidad de Santiago
- › Catedrático de Teoría de la Comunicación Humana de la Universidad Pontificia de Salamanca
- › Presidente del Instituto Ciencia y Sociedad

### D. LUIS M. GIMENEZ-CASSINA BASAGOITI

- › Presidente del Instituto de la Ingeniería de España
- › Subdirector de Ventas IBERIA

### D. JAVIER FERNÁNDEZ DEL MORAL

- › Catedrático de Universidad
- › Director Centro Universitario Villanueva

### D. ORLANDO DOMÍNGUEZ

- › Director del Proyecto Universidad Empresa (PUE)



### D. ROSENDO ÁLVAREZ III

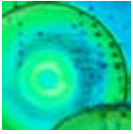
- › Vicepresidente Ejecutivo del Parque Cibernético de Santo Domingo (República Dominicana)

### D. EDUARDO MARTÍNEZ

- › Director revista científica Tendencias21.net

### D. OSCAR RUIZ CUENCA

- › Director General Instituto Novatech



**Instituto Novatech – Grupo Novatech**  
Madrid • Murcia • Alicante • Granada

Central: C/ Gaztambide, 89 (Esq. San Francisco de Sales, 12)  
Tel: 913 952 875 • Móvil: 626 598 625 • Fax: 913 990 019  
info@institutonovatech.com • www.institutonovatech.com