

BLOQUE “JAVA AVANZADO” → IFCT078PO-PATRONES SOFTWARE

- **ACTITUDES BÁSICAS NECESARIAS** (además de las actitudes de los bloques anteriores, añadir:)
 - Cuando ya has resuelto un problema, ¿prefieres buscar otra solución mejor o más eficiente que la anterior, o con la que tenías ya te vale?
 - ¿Te gusta buscar tu perfeccionamiento continuo?
- **CONOCIMIENTOS RECOMENDADOS** (además de los conocimientos de los bloques anteriores)
 - ✓ Conocimientos avanzados de Programación Estructurada.
 - ✓ Conocimientos medios de Programación Orientada a Objetos - POO. Conocimientos medios de Java o de algún otro lenguaje de programación orientado a objetos (C++, etc).
 - ✓ Conocimientos mínimos de HTML.
 - ✓ Uso intermedio de los diagramas UML (lenguaje unificado de modelado).
 - ✓ Conocimientos intermedios de bases de datos relacionales (Oracle, o Mysql, o PostgreSQL, ...).
 - ✓ Manejo avanzado de algún sistema de control de versiones (GitHub, GitLab, Subversion, ...).
 - ✓ Los asistentes a este curso deben tener experiencia previa de programación, si bien, adaptaremos su contenido al conocimiento medio que tengan los alumnos interesados que finalmente resulten admitidos.
 - ✓ Manejo fluido de algún entorno integrado de programación (IDE) como Eclipse, Netbeans, etc.
- **CONTENIDOS FORMATIVOS DEL CURSO:**
 - Los Patrones Software son un conjunto de buenas prácticas cuyo objetivo es dar soluciones generalmente aceptadas a problemas comunes en el ámbito de la programación orientada a objetos. Estos patrones fueron marcados en su momento por la denominada "Banda de los Cuatro", en referencia a cuatro de los programadores más reconocidos a nivel mundial.
 - De forma general veremos los 23 patrones iniciales (dado el tiempo con que contamos no es posible ver en profundidad todos los patrones, por lo que en función del perfil de los asistentes se insistirá más en los patrones más adecuados.), e incidiremos en los más habituales, con ejercicios prácticos, especialmente con los siguientes patrones:
 - Abstract Factory
 - Factory Method
 - Singleton
 - Composite
 - Iterator
 - Observer
 - Strategy
 - DAO
 - MVC
 - Haremos un ejercicio final incluyendo varios de los patrones anteriores, como parte de una aplicación real con Java, Spring, Hibernate, HTML/CSS/Bootstrap, con una base de datos de MySQL.
 - Utilizaremos herramientas de mercado en el ámbito de la programación orientada a objetos, como son: lenguaje Java (entorno Eclipse), Base de datos MySQL, una librería de acceso a BBDD como HIBERNATE, un Framework MVC como SPRING, y utilidades típicas de la programación de páginas web como HTML5 y BOOTSTRAP que nos permitan realizar ejemplos de todo lo que aportan los Patrones Software y las herramientas antes indicadas.

✓ **DETALLE DE PATRONES:**

1. INTRODUCCIÓN DE LOS PATRONES DE DISEÑO TOTAL. 1.1. Orígenes de los patrones de diseño. 1.2. Relación entre los patrones y la orientación objetos. 1.3. Definición de patrón. 1.4. Características. 1.5. Elementos de un patrón. 1.6. Clasificación. 1.7. Patrones significativos. 1.8. Frameworks.

2. PATRONES FUNDAMENTALES TOTAL. 2.1. Introducción a los patrones fundamentales. 2.2. Clasificación de los patrones fundamentales. 2.3. Patrón Delegation. 2.4. Patrón Interface. 2.5. Patrón Abstract Superclass. 2.6. Patrón Interface y Abstract Class. 2.7. Patrón Marker Interface. 2.8. Patrón Proxy.

3. PATRONES DE CREACIÓN TOTAL. 3.1. Introducción a los patrones de creación . 3.2. Clasificación de los patrones de creación. 3.3. Patrón Abstract Factory. 3.4. Patrón Builder. 3.5. Patrón Factory Method. 3.6. Patrón Singleton.

4. PATRONES ESTRUCTURALES TOTAL. 4.1. Introducción a los patrones estructurales. 4.2. Clasificación de los patrones estructurales. 4.3. Patrón Adapter. 4.4. Patrón Composite . 4.5. Patrón Decorator. 4.6. Patrón Facade. 4.7. Patrón Proxy.

5. PATRONES DE COMPORTAMIENTO TOTAL. 5.1. Introducción a los patrones de comportamiento. 5.2. Clasificación de los patrones de comportamiento. 5.3. Patrón Command. 5.4. Patrón Iterator. 5.5. Patrón Observer. 5.6. Patrón State. 5.7. Patrón Strategy. 5.8. Patrón Template Method. 5.9. Patrón Visitor