

Técnicas de programación avanzada

Curso 2023-2024

Presentación de la asignatura

Alejandro Alonso Puig (@mundobot)

ÍNDICE

• El profesor	03
• Los alumnos	04
• Contenidos	05
• Sistema de evaluación	06
• Bibliografía	07

El profesor

- Profesional del sector de la robótica y la automatización.
- Llevo trabajando 30 años en empresas como IBM, Honeywell, ABB/ASTI, IXION Industria y Aeroespacio, Infinium Robotics, Kivnon en áreas de Software, Electrónica y Mecánica.
- Fundé mi propia empresa de robótica (Quark Robotics)
- He vivido en Madrid, Barcelona, Burgos y Singapur
- Profesor en UAX y Nebrija de Industria 4.0, Gestión de operaciones y simulación en la Industria, Electrónica Industrial, Gestión de proyectos tecnológicos, Aplicación de las tecnologías de vanguardia en mecánica, Bioinstrumentación y sensores, Programación avanzada y Tecnología eléctrica y electrónica.



<https://www.linkedin.com/in/alejandroalonsopuig/>



<https://automacomp.blogspot.com/>



<https://www.youtube.com/@mundobot>



<https://twitter.com/mundobot>



<https://github.com/aalonsopuig>



<https://www.instagram.com/aalonsopuig/>

Alejandro Alonso Puig



Los Alumnos

- De qué conocimientos partís
- Qué esperáis de la asignatura
- Qué dificultades veis
- Qué inquietudes tenéis,...

Tema 1: Objetos y memoria.

- 1.1 Características básicas del lenguaje. Primer programa. Compilación y Ejecución. IDE.
- 1.2 Sentencias de control. Secuencia, selección e iteración.
- 1.3 Abstracción. Clases, objetos, métodos y atributos.
- 1.4 Sobrecarga de métodos y encapsulamiento.

Tema 2. Otros conceptos fundamentales de la Programación Orientada a Objetos

- 2.1 Herencia. Interfaces y clases abstractas. Agregación.
- 2.2 Polimorfismo.
- 2.3 Gestión de Excepciones.
- 2.4 Genericidad y plantillas.
- 2.5 Utilidades. Entrada y Salida.
- 2.6 Anotaciones.

Recomendación:

Repasar prog., estudiar nuevos conceptos y practicar, lo antes posible. Es la base para la asignatura.

Tema 3. Patrones de Diseño.

- 3.1 Concepto de Patrones de Diseño.
- 3.2 Patrones de creación.
- 3.3 Patrones estructurales.
- 3.4 Patrones de comportamiento.

Tema 4. Programación de Interfaces.

- 4.1 Interfaces Gráficas de Usuario.
- 4.2 Gestión de eventos.

Tema 5. Temas Avanzados.

- 5.1 Concurrencia.
- 5.2 Inversión de Control. Definición y ejemplos. Inyección de dependencias.
- 5.3 Expresiones avanzadas del lenguaje.

Programación (aprox.)

Sesiones (3h)	Contenido	Fechas
1	Intro + 1.1 y 1.2	04/Sept
2	1.3 y 1.4	11/Sept
3	1.3 y 1.4	18/Sept
4	REPASO/Práctica/Ejercicios	25/Sept
5	2.1 y 2.2	02/Oct
P1	Objetos y Memoria	09/Oct
6	2.3, 2.4	09/Oct
7	2.5 + REPASO/Práctica/Ejercicios	16/Oct
P2	Herencia, polimorfismo (y plantillas)	30/Oct
8	4.1	23/Oct
9	4.2	30/Oct
P4	Programación de interfaces	13/Nov
10	3.1, 3.2	06/Nov
11	Rep 3.2, 3.3, 3.4	13/Nov
P3	Patrones de diseño	20/Nov
12	5.1 y 5.2.1	20/Nov
13	5.2.2 y 5.3	27/Nov
14	REPASO/Práctica/Ejercicios	04/Dic
P5	Concurrencia e inversión del control	04/Dic
15	REPASO ASIGNATURA – DUDAS, Ej. Examen, Prácticas	11/Dic

Grupo mañana (Lunes)

11:30-12:50

Pausa

13:00 – 14:10



**Veremos el
bloque 4
antes que el 3.**

Prácticas (Lunes)

8:30-9:50

Pausa

10:00 – 11:10

Prácticas + Trabajo

Sesión	Contenido
P1	Objetos y Memoria
Creación de la primera jerarquía de clases. Abstracción + Encapsulamiento (+setters, getters). Array of super Class + method call (Crear objetos + ejecutar acción)	
P2	Herencia, polimorfismo (y plantillas)
Modificación y mejora de las jerarquía de clases. Introducción de interfaces y clases abstractas. Agregación y composición. Introducción de excepciones, plantillas y @notaciones	
P4	Programación de interfaces
Creación de la GUI	
P3	Patrones de diseño
Optimización del repositorio a través de la introducción de patrones de diseño (obligatorio 1 patrón creacional, estructural o de comportamiento).	
P5	Concurrencia e inversión del control
Introducción de concurrencia y dependencias externas (si procede). Mejora final del programa con expresiones avanzadas. Optimización.	

Trabajo. Entregas obligatorias

P1: Requisitos del programa

P2: Esquema UML del programa
(hecho a mano o a ordenador).
Pueden incluirse bocetos de la APP,
notas, explicaciones (opcional)...

P4: Detalles de la GUI que se
utilizará. Bocetos, estructura,...

P3: Detalle de patrones de diseño
utilizados

P5: Borrador del programa

11-Dic: Programa completo y
memoria del Proyecto.

Evaluación: Continua (35%) + Final (65%)

Convocatoria ordinaria

Sistemas de evaluación	Porcentaje
Evaluación de la participación del alumno	5%
Actividades dirigidas, prácticas y memorias de prácticas, trabajos (obligatorios y voluntarios) y proyectos a realizar.	15%
Prueba escrita parcial	15%
Prueba escrita final	65%

Convocatoria extraordinaria

Sistemas de evaluación	Porcentaje
Actividades dirigidas, prácticas y memorias de prácticas, trabajos (obligatorios y voluntarios) y proyectos a realizar.	10%
Prueba escrita final	90%

Calificación mínima

La ponderación tanto del examen parcial como de los conceptos de participación y trabajos escritos/prácticas, solo se aplicará si el alumno obtiene al menos un 5 en el examen final. Esta ponderación también se aplica solo en el caso de que el alumno obtenga al menos un 5 en este examen final extraordinario.

La no superación de las prácticas supone el suspenso automático de la asignatura en la convocatoria ordinaria y extraordinaria.

Las prácticas que no hayan sido aprobadas pueden, en su caso, ser entregadas de nuevo para ser evaluadas en la convocatoria extraordinaria, previa consulta al profesor y siempre antes del examen de la convocatoria ordinaria.

El examen parcial no libera materia.

Asistencia

El alumno que, injustificadamente, deje de asistir a más de un 25% de las clases presenciales podrá verse privado del derecho a examinarse en la convocatoria ordinaria.

El plagio es un delito. En caso de detectarse este tipo de prácticas, se considerará Falta Grave y se podrá aplicar la sanción prevista en el Reglamento del Alumno.

[← Back to courses](#)


Java

With our interactive Java course, you'll learn object-oriented Java programming and have the ability to write clear and valid code in almost no time at all.



Basic Concepts

10/10



Conditionals and Loops

10/10



Arrays

0/0



Classes and Objects

15/15



More on Classes

15/15



Exceptions, Lists, Threads & Files

15/15



Certification

We are happy to present your certificate to you for completing this course.

Lessons Completed

Projects Completed

[Claim Certificate](#)

Subida de nota (optativo), para redondear hacia arriba

Completar en [Sololearn.com](https://sololearn.com) el curso de:

- Java

Entregar certificado obtenido al finalizar el curso y dar nombre de usuario para comprobación.



- ❖ Deitel, H. M. & Deitel, P. J.(2008). **Java: como programar**. Pearson education. Séptima edición.
- ❖ **Sumérgete en los patrones de diseño**. V2021-1.7. Alexander Shvets. <https://refactoring.guru/es/design-patterns/book>

Versión online: <https://refactoring.guru/es/design-patterns/catalog>

- ❖ The Java tutorials. <https://docs.oracle.com/javase/tutorial/>

- ❖ Páginas de apoyo para la sintaxis, bibliotecas, etc.:

- ❖ <https://www.sololearn.com>

- ❖ <https://www.w3schools.com/java/default.asp>

- ❖ Tutorial 2D game from scratch (GUI): <https://youtu.be/dEKs-3GhVKQ>

Code: <https://github.com/CodeNMore/New-Beginner-Java-Game-Programming-Src>

***A programar se aprende
programando.***

Importante: ser capaz de
encontrar la documentación y
librerías necesarias

Técnicas de Programación Avanzada

```
exit(); //Gracias!
```