

## 804227 - PROG2VJ - Programación II

Unidad responsable: 804 - CITM - Centro de la Imagen y la Tecnología Multimedia  
Unidad que imparte: 804 - CITM - Centro de la Imagen y la Tecnología Multimedia  
Curso: 2018  
Titulación: GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS (Plan 2014). (Unidad docente Obligatoria)  
GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS (Plan 2014). (Unidad docente Obligatoria)  
Créditos ECTS: 6 Idiomas docencia: Catalán, Castellano, Inglés

### Profesorado

Responsable: Lluch Ariet, Magí  
Otros: Díaz García, Jesús

### Competencias de la titulación a las cuales contribuye la asignatura

#### Específicas:

3. Utilizar lenguajes de programación, patrones algorítmicos, estructuras de datos, herramientas visuales de programación, motores de juego y librerías para el desarrollo y prototipado de videojuegos, de cualquier género y para cualquier plataforma y dispositivo móvil.

#### Genéricas:

1. Interpretar los fundamentos del uso y programación de los computadores, los sistemas operativos, las bases de datos y, en general, los programas informáticos con aplicación en ingeniería.  
2. Interpretar y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para el tratamiento automático de la información por medio de sistemas computacionales y su aplicación para la resolución de problemas propios de la ingeniería.

#### Transversales:

4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicarse de forma oral y escrita con otras personas sobre los resultados del aprendizaje, de la elaboración del pensamiento y de la toma de decisiones; participar en debates sobre temas de la propia especialidad.  
5. TERCERA LENGUA: Conocer una tercera lengua, preferentemente el inglés, con un nivel adecuado oral y escrito y en consonancia con las necesidades que tendrán los titulados y tituladas.  
7. USO SOLVENTE DE LOS RECURSOS DE INFORMACIÓN: Gestionar la adquisición, la estructuración, el análisis y la visualización de datos e información en el ámbito de especialidad, y valorar de forma crítica los resultados de dicha gestión.

### Metodologías docentes

Las clases de aprendizaje dirigido estructuran en sesiones de dos horas. Durante parte de las sesiones, el profesor expone los conceptos teóricos y lo ejemplifica mediante ejemplos que se resuelven, en lo posible, de forma participativa por parte de los estudiantes. Otra parte de la sesión se dedica a que los estudiantes practiquen los conceptos introducidos resolviendo una serie de ejercicios propuestos por el profesorado y, cuando proceda, también se dedicará tiempo para la resolución de dudas y problemas con los que se hayan encontrado durante la realización de los ejercicios.

Se hará un uso intensivo del campus virtual, tanto para publicar el material de la asignatura (apuntes, enunciados de problemas, soluciones propuestas, recopilación de links, etc.) como mecanismo de comunicación para publicar avisos, pedir las revisiones de las diferentes pruebas, etc.

### Objetivos de aprendizaje de la asignatura

· Consolidar los conceptos relativos a la utilización de punteros y gestión de memoria dinámica.

## 804227 - PROG2VJ - Programación II

- Asimilar los conceptos propios de la programación orientada a objetos y ser capaz de utilizarlos adecuadamente.
- Conocer las principales estructuras de datos y saber utilizarlas: vectores, estructuras secuenciales (pilas, colas y listas), árboles y grafos.
- Aprender las bases de la programación recursiva y generación de fractales.
- Conocer, analizar y saber implementar y adaptar los principales algoritmos de búsqueda y ordenación.

### Horas totales de dedicación del estudiantado

Dedicación total: 150h	Horas grupo grande:	24h	16.00%
	Horas grupo mediano:	16h	10.67%
	Horas grupo pequeño:	0h	0.00%
	Horas actividades dirigidas:	20h	13.33%
	Horas aprendizaje autónomo:	90h	60.00%

## 804227 - PROG2VJ - Programación II

### Contenidos

<p>1. Punteros y gestión de memoria</p>	<p>Dedicación: 8h Grupo grande/Teoría: 4h Actividades dirigidas: 4h</p>
<p>Descripción: Descripción:</p> <ul style="list-style-type: none"> <li>· Punteros</li> <li>· Paso de parámetros (por valor y referencia)</li> <li>· Gestión de la memoria dinámica: reservar y liberar memoria</li> </ul>	
<p>2. Programación orientada a objetos</p>	<p>Dedicación: 18h Grupo grande/Teoría: 8h Actividades dirigidas: 10h</p>
<p>Descripción: Descripción:</p> <ul style="list-style-type: none"> <li>· Clases, encapsulación y diseño modular</li> <li>· Sobrecarga de operadores</li> <li>· Herencia</li> <li>· Polimorfismo</li> <li>· Friendship y templates</li> </ul>	
<p>3. Estructuras de datos</p>	<p>Dedicación: 16h Grupo grande/Teoría: 8h Actividades dirigidas: 8h</p>
<p>Descripción: Descripción:</p> <ul style="list-style-type: none"> <li>· Strings</li> <li>· Pilas y colas</li> <li>· Listas</li> <li>· Vectores dinámicos</li> <li>· Árboles</li> </ul>	

## 804227 - PROG2VJ - Programación II

<p>4. Recursividad y Fractales</p>	<p>Dedicación: 8h Grupo grande/Teoría: 4h Actividades dirigidas: 4h</p>
<p>Descripción: Estructura de los algoritmos recursivos Recursividad en series numéricas Las Torres de Hanoi Fractales: El conjunto de Cantor y el triángulo de Sierpinski</p>	
<p>5. Algoritmos de ordenación</p>	<p>Dedicación: 10h Grupo grande/Teoría: 6h Actividades dirigidas: 4h</p>
<p>Descripción: Descripción: · Bubblesort · Quicksort · Binary Tree · Heap Sort · Merge Sort</p>	

## 804227 - PROG2VJ - Programación II

### Planificación de actividades

<p>Actividades sobre punteros y memoria</p>	<p>Dedicación: 8h Aprendizaje autónomo: 4h Grupo mediano/Prácticas: 4h</p>
<p><b>Descripción:</b>          Dos conjuntos de preguntas y pequeños ejercicios permitirán practicar con los conceptos de apuntadores y gestión de la memoria.          El primer conjunto de ejercicios corresponde a la implementación y manejo de variables básicas y apuntadores hacia ellas, modificando tanto su dirección como el contenido de la memoria donde apuntan.          El segundo conjunto de ejercicios incorporará estructuras, definiendo variables que sean instancias de las estructuras o apuntadores a ellas, permitiendo así empezar a trabajar con operadores de acceso a los miembros de la estructura, que después serán utilizados por los objetos y sus miembros</p> <p><b>Material de soporte:</b>          Enunciado de los ejercicios y MS Visual C++</p> <p><b>Descripción de la entrega esperada y vínculos con la evaluación:</b>          El resultado de la actividad corresponde a un conjunto de ficheros de código C++ que se tendrán que enviar al espacio de entregas del campus y eventualmente presentarlos delante del grupo del aula. Estas entregas y presentaciones servirán para evaluar el 2% de la nota correspondiente a la participación del estudiante.</p> <p><b>Objetivos específicos:</b>          Practicar con los contenidos teóricos expuestos y coger práctica con la implementación de programas en C++</p>	
<p>Actividades de Fundamentos de Programación Orientada a Objetos</p>	<p>Dedicación: 18h Grupo mediano/Prácticas: 8h Aprendizaje autónomo: 10h</p>
<p><b>Descripción:</b>          Conjunto de ejercicios de programación para practicar los conceptos de encapsulamiento, herencia y polimorfismo. Entre otros, se trabajarán ejercicios para representar y manipular burbujas, cápsulas, vehículos, Pokemons o vectores.</p> <p><b>Material de soporte:</b>          Enunciado de los ejercicios y MS Visual C++</p> <p><b>Descripción de la entrega esperada y vínculos con la evaluación:</b>          El resultado de la actividad corresponde a un conjunto de ficheros de código C++ que se tendrán que enviar al espacio de entregas del campus y eventualmente presentarlos delante del grupo del aula. Estas entregas y presentaciones servirán para evaluar el 2% de la nota correspondiente a la participación del estudiante.</p> <p><b>Objetivos específicos:</b>          Practicar con los contenidos teóricos expuestos y coger práctica con la implementación de programas en C++</p>	
<p>Actividades de Estructuras de Datos</p>	<p>Dedicación: 18h Aprendizaje autónomo: 8h Grupo mediano/Prácticas: 10h</p>

## 804227 - PROG2VJ - Programación II

### Descripción:

Cuestionario tipo concurso en el aula con preguntas para repasar los conceptos expuestos  
Conjunto de ejercicios de programación para implementar los objetos asociados a las estructuras básicas y practicar su uso con algunos ejemplos. Se tratarán: i) Pilas, ii) Colas, iii) Listas, iv) Arrays dinámicos, v) Matrices y vi) Árboles.

Estos ejercicios servirán para reforzar el aprendizaje de programación orientada a objetos.

### Material de soporte:

Enunciado de los ejercicios y MS Visual C++

### Descripción de la entrega esperada y vínculos con la evaluación:

El resultado de la actividad corresponde a un conjunto de ficheros de código C++ que se tendrán que enviar al espacio de entregas del campus y eventualmente presentarlos delante del grupo del aula. Estas entregas y presentaciones servirán para evaluar el 2% de la nota correspondiente a la participación del estudiante.

### Objetivos específicos:

Practicar con los contenidos teóricos expuestos y coger práctica con la implementación de programas en C++  
Ser capaz de identificar las estructuras de datos adecuadas para resolver diferentes problemas y situaciones

### Actividades de recursividad y fractales

Dedicación: 8h

Aprendizaje autónomo: 4h

Grupo mediano/Prácticas: 4h

### Descripción:

Practicar y resolver algunos problemas y juegos intrínsecamente recursivos  
Conjunto de ejercicios de programación para implementar las soluciones a los problemas y juegos practicados, incluyendo series numéricas, las torres de Hanoi y fractales

### Material de soporte:

Enunciado de los ejercicios y MS Visual C++

### Descripción de la entrega esperada y vínculos con la evaluación:

El resultado de la actividad corresponde a un conjunto de ficheros de código C++ que se tendrán que enviar al espacio de entregas del campus y eventualmente presentarlos delante del grupo del aula. Estas entregas y presentaciones servirán para evaluar el 2% de la nota correspondiente a la participación del estudiante.

### Objetivos específicos:

Asimilar el concepto de programación recursiva  
Ser capaz de implementar algoritmos recursivos

### Actividades de algoritmos de ordenación

Dedicación: 8h

Aprendizaje autónomo: 4h

Grupo mediano/Prácticas: 4h

### Descripción:

Practicar diferentes métodos de ordenación de series numéricas  
Conjunto de ejercicios de programación para implementar y utilizar los algoritmos de ordenación: i) burbujas, ii) heapsort y iii) quicksort  
Incorporar las estructuras de datos necesarias para cada método de ordenación

### Material de soporte:

Enunciado de los ejercicios y MS Visual C++

## 804227 - PROG2VJ - Programación II

Descripción de la entrega esperada y vínculos con la evaluación:

El resultado de la actividad corresponde a un conjunto de ficheros de código C++ que se tendrán que enviar al espacio de entregas del campus y eventualmente presentarlos delante del grupo del aula. Estas entregas y presentaciones servirán para evaluar el 2% de la nota correspondiente a la participación del estudiante.

Objetivos específicos:

Identificar y utilizar las estructuras de datos adecuadas para cada método de ordenación

Saber aplicar y entender el funcionamiento de los algoritmos recursivos en los métodos de ordenación que los incorporen

Ser capaz de utilizar los algoritmos de ordenación para ordenar diferentes objetos en diferentes escenarios

### Sistema de calificación

Distinguimos tres tipos de actividades evaluables:

- Cuatro prácticas de evaluación sobre temas concretos de la asignatura que los alumnos resolverán en horario de clase (ET)
- Un examen parcial realizado durante la semana prevista en el calendario académico (EP)
- El examen final de la asignatura. (EF)

La nota de la asignatura (NF) se calculará utilizando la siguiente fórmula:

$$NF = 0.2 * EP + 0.4 * TE + 0.4 * EF + 0.1 * PART$$

siendo EP la nota del examen parcial, ET las notas de los 4 ejercicios puntuables a realizar durante el curso (cada uno vale un 10% de la nota final) y EF la nota del examen final. También se evaluará la participación (PART) del alumno a partir de sus intervenciones y del interés de aprendizaje demostrado en clase.

Los alumnos que no superen la asignatura tendrán la posibilidad de presentarse a la prueba de Reevaluación (se reevaluará el 90% correspondiente a los exámenes y ejercicios prácticos realizados en clase. La nota final de la asignatura después de la reevaluación, será como máximo un 5).

### Normas de realización de las actividades

La realización de las diferentes pruebas se realizarán exclusivamente a través de los mecanismos establecidos, en las fechas y horarios indicados.

### Bibliografía

Básica:

Stroustrup, B. A tour of C++. Addison-Wesley, 2013. ISBN 9780321958310.

Kernighan, Brian W; Ritchie, Dennis M. The C programming language. 2nd ed. New Jersey: Prentice Hall, 1988. ISBN 0131103628.

Franch Gutiérrez, Xavier [et al.]. Fonaments de programació: problemes resolts en C++ [en línia]. Barcelona: Edicions UPC, 2006 [Consulta: 10/12/2018]. Disponible a: <<http://hdl.handle.net/2099.3/36692>>. ISBN 9788483018828.

Otros recursos:

C++ Reference: <http://www.cplusplus.com/reference/>

C and C++ Programming: <http://www.cprogramming.com/>