



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

GUÍA DOCENTE DE PROGRAMACIÓN II

La presente guía docente corresponde a la asignatura Programación II (PROG2), aprobada para el curso lectivo 2017-2018 en Junta de Centro y publicada en su versión definitiva en la página web de la Escuela Politécnica Superior. La guía docente de PROG2 aprobada y publicada antes del periodo de matrícula tiene el carácter de contrato con el estudiante.



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

ASIGNATURA

PROGRAMACIÓN II (PROG2)

1.1. Código

18468 del Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

1.2. Materia

Informática

1.3. Tipo

Formación básica

1.4. Nivel

Grado

1.5. Curso

1º

1.6. Semestre

2º

1.7. Número de créditos

6 créditos ECTS

1.8. Requisitos previos

Programación II forma parte de la *Materia Informática* del módulo de *Formación Básica* del plan de estudios. Esta Materia se desglosa en tres asignaturas semestrales que se complementan entre sí: *Programación I*, *Programación II* y *Fundamentos de Sistemas Informáticos*. La primera de ellas se imparte en el primer semestre, mientras que la segunda lo hace en el segundo cuatrimestre de primer curso.

Aunque no existen requisitos académicos para cursar la asignatura *Programación II*, es necesario que el alumno disponga de unos conocimientos básicos relativos a los



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

fundamentos de programación y la metodología del diseño de aplicaciones software utilizando el lenguaje C. Esto implica haber superado con éxito la asignatura de *Programación I*.

En concreto, el alumno, antes de cursar la asignatura, debería:

- Ser capaz de diseñar algoritmos y expresarlos en un pseudocódigo independiente de un lenguaje de programación específico.
- Conocer el funcionamiento de la recursividad y ser capaz de diseñar algoritmos recursivos.
- Ser capaz de transformar un algoritmo expresado en pseudocódigo en un programa C.
- Ser capaz de utilizar de manera eficaz y fluida diferentes entornos de programación, incluyendo las herramientas de edición, compilación, ensamblaje y depurado de código.

Se recomienda para garantizar la asimilación de los contenidos y la adquisición de habilidades la lectura crítica de los textos de la bibliografía, el uso del material electrónico de esta asignatura disponible en la plataforma Moodle (<http://uam-virtual.es>) y la búsqueda activa de material complementario en la red. Asimismo, es recomendable que el estudiante disponga de un dominio del idioma inglés que le permita leer la bibliografía de consulta. Finalmente, se requiere iniciativa personal y constancia para el diseño/escritura de programas y la resolución de los ejercicios propuestos durante el curso junto con predisposición y empatía para el trabajo colaborativo en grupo.

1.9. Requisitos mínimos de asistencia a las sesiones presenciales

Esta asignatura dispone de dos métodos de evaluación para la parte teórica de la asignatura: continua y no continua.

La asignatura solo dispone de un método de evaluación para la parte práctica.

Se supone que todos los estudiantes se acogen al método de evaluación continua a no ser que comuniquen explícitamente lo contrario al profesor antes del día en que tenga lugar la prueba final.

Los detalles de los métodos de evaluación se encuentran en la sección 4 de esta guía.

La asistencia a clase de teoría es muy recomendable. La asistencia a clase de prácticas es obligatoria al menos en un 85%.



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

1.10. Datos del equipo docente

Nota: se debe añadir @uam.es a todas las direcciones de correo electrónico.

Profesores de teoría:

Dra. Estrella Pulido Cañabate (Coordinadora)
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Módulo: B-413 Edificio B - 4ª Planta
Teléfono: +34 91 497 2289
Correo electrónico: estrella.pulido
Página web: <http://www.ii.uam.es/~epulido>
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Profesores de prácticas:

Dr. Gonzalo Martínez Muñoz
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Módulo: B-422 Edificio B - 4ª Planta
Teléfono: +34 91 497 7528
Correo electrónico: gonzalo.martinez
Página web: <http://www.ii.uam.es/~gonzalo>
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Jesús Sánchez Cuadrado
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Módulo: B-318 Edificio B - 3ª Planta
Teléfono: +34 91 497 2251
Correo electrónico: jesus.sanchez.cuadrado
Página web:
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

1.11. Objetivos del curso

La asignatura PROG2 tiene como objeto (1) proporcionar unos conocimientos básicos sobre la especificación, implementación y uso de los tipos abstractos de datos e (2) iniciar en los fundamentos del diseño, análisis e implementación de algoritmos.



Las **competencias generales** que el estudiante adquiere con la asignatura PROG2 son:

- **DD3:** Capacidad de reunir e interpretar datos relevantes, en el ámbito de la ingeniería de tecnologías y servicios de telecomunicación, para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
- **ITT3:** Conocimiento de materias básicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías, así como que le dote de una gran versatilidad para adaptarse a nuevas situaciones.

La **competencia específica** que el estudiante adquiere con la asignatura PROG2 es:

- **FB2:** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Los **resultados del aprendizaje** que el estudiante adquiere con la asignatura PROG2 son:

- Comprensión de tipos abstractos de datos así como su implementación en lenguajes de programación de alto nivel.
- Comprensión de los principales algoritmos así como de las técnicas necesarias para la estimación de su complejidad.

Al final de cada tema el estudiante deberá ser capaz de:

OBJETIVOS ESPECIFICOS POR TEMA	
TEMA 1.- Introducción	
1.1.	Definir el concepto de tipo abstracto de datos (TAD)
1.2.	Enumerar las ventajas de utilizar tipos abstractos de datos
1.3.	Describir los elementos que forman parte de la especificación de un TAD
1.4.	Definir los conceptos de algoritmo y programa
1.5.	Describir las propiedades de un algoritmo
TEMA 2.- Listas	
2.1.	Definir el concepto de lista
2.2.	Justificar el uso de listas enlazadas para la implementación del TAD Lista
2.3.	Definir el concepto de lista enlazada
2.4.	Justificar el uso del nodo cabecera en listas enlazadas
2.5.	Codificar en lenguaje C la definición del TAD Lista
2.6.	Codificar en lenguaje C las operaciones del TAD Lista
2.7.	Definir el concepto de lista doblemente enlazada
2.8.	Describir las ventajas e inconvenientes de las listas doblemente enlazadas
2.9.	Definir el concepto de lista circular simplemente y doblemente enlazada
2.10.	Codificar en lenguaje C las operaciones de suma y multiplicación de polinomios utilizando el TAD Lista



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

TEMA 3.- Pilas	
3.1.	Definir el concepto de pila
3.2.	Describir las operaciones de apilar, desapilar y cima
3.3.	Codificar en lenguaje C la definición del TAD Pila
3.4.	Codificar en lenguaje C las operaciones del TAD Pila
3.5.	Codificar en lenguaje C la función de evaluación de una expresión en notación sufija utilizando el TAD Pila
TEMA 4.- Colas	
4.1.	Definir el concepto de cola
4.2.	Describir las operaciones de encolar, desencolar y primero
4.3.	Codificar en lenguaje C la definición del TAD Cola
4.4.	Codificar en lenguaje C las operaciones del TAD Cola
4.5.	Definir el concepto de cola circular
4.6.	Codificar en lenguaje C las operaciones del TAD Cola Circular
4.7.	Definir el concepto de cola de doble entrada o bicola
4.8.	Enumerar y describir los tipos de bicola existentes
4.9.	Codificar en lenguaje C las operaciones del TAD Bicola
4.10.	Definir el concepto de cola de prioridad
4.11.	Describir cómo funciona y cómo se implementa una cola de prioridad
4.12.	Enumerar las operaciones del TAD Cola de Prioridad
4.13.	Codificar en lenguaje C la función que evalúa si una frase es un palíndromo utilizando los TAD Pila y Cola
TEMA 5.- Árboles	
5.1.	Definir el concepto de árbol
5.2.	Definir la terminología básica relacionada con los árboles (nodo, raíz, arco, hoja, grado, hijo, padre, camino, nivel, altura, ...)
5.3.	Codificar en lenguaje C la definición del TAD Árbol
5.4.	Definir el concepto de árbol binario
5.5.	Definir el concepto de recorrido de un árbol binario
5.6.	Describir y aplicar a ejemplos los diferentes recorridos en árboles binarios
5.7.	Definir el concepto de árbol binario de búsqueda
5.8.	Aplicar las operaciones de búsqueda, inserción y borrado en árboles binarios de búsqueda
5.9.	Definir el concepto de árbol AVL
5.10.	Aplicar las operaciones de inserción y borrado en árboles AVL
TEMA 6.- Grafos	
6.1.	Definir el concepto de grafo
6.2.	Definir la terminología básica relacionada con los grafos (dirigido, no dirigido, adyacente, camino, ciclo, grado ...)
6.3.	Representar un grafo mediante una matriz de adyacencia
6.4.	Describir los problemas que plantea representar un grafo mediante una matriz de adyacencia
6.5.	Representar un grafo mediante una lista de adyacencia
6.6.	Describir los problemas que plantea representar un grafo mediante una lista de



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

	adyacencia
6.7.	Describir y aplicar a ejemplos los diferentes recorridos en grafos
6.8.	Aplicar el algoritmo de búsqueda de caminos más cortos en un grafo no ponderado
6.9.	Aplicar el algoritmo de Dijkstra a un grafo ponderado
TEMA 7.- Algoritmos de ordenación	
7.1.	Describir el funcionamiento general de cada uno de los algoritmos de ordenación
7.2.	Aplicar a ejemplos concretos cada uno de los algoritmos de ordenación
7.3.	Escribir el pseudocódigo de cada uno de los algoritmos de ordenación
TEMA 8.- Algoritmos de búsqueda	
8.1.	Describir el funcionamiento general de cada uno de los algoritmos de búsqueda
8.2.	Aplicar a ejemplos concretos cada uno de los algoritmos de búsqueda
8.3.	Escribir el pseudocódigo de cada uno de los algoritmos de búsqueda
TEMA 9.- Análisis de complejidad	
9.1.	Describir en qué consiste el análisis de algoritmos y qué tipos de análisis existen
9.2.	Definir el concepto de notación O
9.3.	Enumerar las propiedades de la notación O
9.4.	Analizar la complejidad de instrucciones básicas de programación
9.5.	Analizar la complejidad de algoritmos específicos

1.12. Contenidos del programa

Programa Sintético

TEMA 1. Punteros y memoria dinámica en C
TEMA 2. Introducción
TEMA 3. Listas
TEMA 4. Pilas
TEMA 5. Colas
TEMA 6. Árboles
TEMA 7. Grafos
TEMA 8. Algoritmos de ordenación
TEMA 9. Algoritmos de búsqueda
TEMA 10. Análisis de complejidad

Programa Detallado

1. Punteros y memoria dinámica en C
 - 1.1. Punteros
 - 1.2. Memoria dinámica
2. Introducción
 - 2.1. Tipos de datos
 - 2.2. Tipos abstractos de datos



- 2.3. Algoritmos
- 3. **Listas**
 - 3.1. Conceptos básicos
 - 3.2. Implementación
 - 3.3. Listas enlazadas
 - 3.3.1. Definiciones para el TAD Lista
 - 3.3.2. Operaciones: crearLista, esVacia, insertar, imprimirLista, buscarElemento, buscarAnterior, esUltimo, borrar, borrarLista
 - 3.4. Listas doblemente enlazadas
 - 3.5. Listas circulares
 - 3.6. Aplicación: polinomios
- 4. **Pilas**
 - 4.1. Conceptos generales
 - 4.2. Definiciones para el TAD Pila
 - 4.3. Operaciones: crearPila, esVacia, apilar, desapilar, cima, borrarPila, imprimirPila
 - 4.4. Aplicación: notación sufija
- 5. **Colas**
 - 5.1. Conceptos generales
 - 5.2. Definiciones para el TAD Cola
 - 5.3. Operaciones: crearCola, esVacia, encolar, desencolar, primero, borrarCola, imprimirCola
 - 5.4. Colas circulares
 - 5.5. Colas de doble entrada o bicolas
 - 5.6. Colas de prioridad
 - 5.7. Aplicación: palíndromos
- 6. **Árboles**
 - 6.1. Conceptos básicos
 - 6.2. Implementación
 - 6.3. Árboles binarios
 - 6.4. Recorridos
 - 6.5. Árboles binarios de búsqueda
 - 6.6. Árboles AVL
- 7. **Grafos**
 - 7.1. Conceptos básicos
 - 7.2. Representación
 - 7.3. Recorridos
 - 7.4. Caminos más cortos en grafos no ponderados
 - 7.5. Algoritmo de Dijkstra
- 8. **Algoritmos de ordenación**
 - 8.1. Algoritmo de selección
 - 8.2. Algoritmo de inserción
 - 8.3. Algoritmo de burbuja
 - 8.4. Heapsort
 - 8.5. Mergesort



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

- 8.6. Quicksort
- 9. Algoritmos de búsqueda
 - 9.1. Búsqueda lineal
 - 9.2. Búsqueda binaria
 - 9.3. Tablas de dispersión
- 10. Análisis de complejidad
 - 10.1. Conceptos generales
 - 10.2. Notación O
 - 10.3. Análisis de instrucciones simples
 - 10.4. Ejemplos de análisis

1.13. Referencias de consulta

Bibliografía:

1. Data Structures and algorithm analysis in C, Weiss Mark Allen, Addison Wesley--Longman. 1998. INF/681.3.01/WEI
2. Algoritmos y estructuras de datos: una perspectiva en C, Joyanes Aguilar L., Zahonero Martínez I., Mc Graw Hill 2003. INF/510.5/JOY
3. Estructuras de datos con C y C++, Langsam, Augenstein, Tenenbaum, Prentice Hall Hispanoamericana. 1997. INF/681.3.01/LAN
4. Algoritmos + Estructuras de Datos = Programas. N. Wirth, Ediciones del Castillo. 1986. INF/681.3.01/WIR
5. The C programming language. Brian W. Kernighan, Dennis M. Ritchie. Prentice Hall, 1988. INF/681.3.062-C/KER
6. Estructura de Datos y Algoritmos. M. A. Weiss, Addison Wesley. 1995. INF/681.3.01/WEI
7. Estructuras de datos y algoritmos, Aho, Hopcroft, Ullman, Addison--Wesley 1998. INF/681.3.01/AHO

Nota: no se recomienda a los estudiantes comprar ningún libro hasta no haber comparado su contenido con el programa y revisado previamente en la biblioteca.

Material electrónico de trabajo: los documentos electrónicos de trabajo se publican en la sección de PROG2 en plataforma Moodle (<http://uam-virtual.es>)

2. Métodos docentes

La metodología utilizada en el desarrollo de la actividad docente incluye los siguientes tipos de actividades:

➤ Clases de teoría:

Actividad del profesor



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

Propuesta de programas y ejercicios. Supervisión y asesoramiento en la resolución de los problemas por parte del alumno y, al finalizar, análisis del resultado, explicación de los conceptos que presenten más dificultades y generalización a otros tipos de problemas. Se utilizará la pizarra y el proyector del aula para visualizar los algoritmos o programas propuestos.

Actividad del estudiante:

Actividad presencial: Participación activa en clase respondiendo a las cuestiones planteadas. Resolución de los ejercicios propuestos, diseño y escritura de pequeños programas durante el desarrollo de las clases.

Actividad no presencial: revisión del material de apoyo, estudio de la materia y realización de los cuestionarios y ejercicios planteados en clase o a través de las plataformas online utilizadas.

➤ **Tutorías en aula:**

Actividad del profesor:

Tutorización a toda la clase o en grupos de alumnos reducidos (8-10) con el objetivo de resolver dudas comunes planteadas por los alumnos a nivel individual o en grupo, surgidas a partir de cuestiones/ejercicios/programas señalados en clase para tal fin y orientarlos en la realización de los mismos.

Actividad del estudiante:

Actividad presencial: Planteamiento de dudas individuales o en grupo y enfoque de posibles soluciones a las tareas planteadas.

Actividad no presencial: Estudio de las tareas marcadas y debate de las soluciones planteadas en el seno del grupo.

➤ **Prácticas:**

Actividad del profesor:

Asignar una práctica/proyecto a cada grupo de trabajo y explicar la práctica asignada a cada grupo de trabajo al comienzo de la sesión de prácticas. Supervisar el trabajo de los grupos de trabajo en el laboratorio. Suministrar el guión de prácticas a completar en el laboratorio.

Se utilizan el método expositivo tanto en tutorías como en el laboratorio con cada grupo de trabajo. Los medios utilizados son los entornos de programación, editores, compiladores y los ordenadores del propio laboratorio para la ejecución y análisis de los programas realizados.

Actividad del estudiante:

Actividad presencial: Planteamiento inicial, previo al desarrollo de la práctica, sobre información contenida en el enunciado. Debate en el seno del grupo sobre el planteamiento de la solución óptima. Al finalizar la práctica se entrega un breve informe con el programa desarrollado y, además, se debe ejecutar con el profesor presente, quien hará las preguntas oportunas a cada miembro del grupo para calificar de forma individual la práctica.



Actividad no presencial: Profundizar en el enunciado de la práctica y plantear el diagrama de flujo óptimo para la resolución de la misma. Redacción del informe de la práctica.

3. Tiempo de trabajo del estudiante

		Nº de horas	Porcentaje
Presencial	Clases teórico-prácticas	70 h (47%)	82 h (55%)
	Tutorías	6 h (4%)	
	Realización de pruebas finales	6 h (4%)	
No presencial	Estudio semanal	36 h (24%)	68 h (45%)
	Preparación de pruebas finales	32 h (21%)	
Carga total de horas de trabajo: 25 horas x 6 ECTS		150 h	

4. Métodos de evaluación y porcentaje en la calificación final

- ✓ Ambas partes, teoría y prácticas se puntúan sobre 10 puntos.
- ✓ La nota final de la asignatura se obtiene de las notas de teoría y prácticas por medio de la ecuación:

$$\text{Calificación: } 0.4 * \text{Prácticas} + 0.6 * \text{Teoría}$$

- ✓ Para aprobar la asignatura es obligatorio obtener una nota mayor o igual a 5 puntos en la parte de teoría y una nota mayor o igual a 5 en las prácticas. En caso contrario, la nota final en actas será la siguiente:

$$\text{Calificación: } (0,4 * \text{Mín}(5, \text{Prácticas}) + 0,6 * \text{Mín}(5, \text{Teoría}))$$

- ✓ La nota correspondiente a la parte de Teoría para el método de evaluación continua es la que resulta de la siguiente fórmula (donde p1, p2 y p3 son las calificaciones obtenidas en las tres pruebas intermedias):

$$\text{Calificación_continua} = 0,25 * p1 + 0,35 * p2 + 0,35 * p3 + 0,05 * \text{ejercicios_clase}$$

- ✓ La tercera prueba intermedia coincidirá en fecha con la prueba final.
- ✓ La nota correspondiente a la parte de Teoría para el método de evaluación no continua corresponde únicamente a la prueba final.

- ✓ La nota correspondiente a la parte de Prácticas es la que resulta de aplicar la siguiente fórmula

$$\text{Calificación_prácticas} = 0,05 * p1 + 0,15 * p2 + 0,4 * p3 + 0,4 * p4$$



Asignatura: Programación II
Código: 18468
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

donde p1, p2, p3 y p4 son las calificaciones de cada una de las prácticas que se calculan a partir de las notas obtenidas en la entrega de la práctica, en los controles semanales sobre el progreso de cada práctica y en la prueba de evaluación sobre la misma. Para aplicar la fórmula anterior, es necesario

- ✓ obtener una calificación mínima de 3 puntos en cada una de las prácticas
- ✓ superar al menos 3 de los controles semanales para cada una de las prácticas

Convocatoria extraordinaria.

- Se aplican las mismas fórmulas y condiciones que en la convocatoria ordinaria con dos salvedades:
 - La calificación de teoría corresponde a una única prueba similar a la prueba final de la convocatoria ordinaria.
 - La calificación de la entrega de prácticas corresponde a la obtenida en la realización de una única práctica y a una prueba sobre la misma.
- Tanto la nota de teoría como la de prácticas se conservan (convalidan) sólo para la convocatoria extraordinaria del mismo curso académico.

ATENCIÓN: Cualquier copia descubierta que se haya realizado a lo largo del curso, tanto en cualquiera de las actividades de teoría desarrolladas, como en cualquiera de los apartados de las prácticas, serán penalizadas con rigurosidad. La penalización por copia implica la aplicación de la normativa interna de la EPS, que supone suspender la convocatoria actual.

5. Cronograma

Semana	Contenido	Horas presenciales	Horas no presenciales
1	- Presentación y motivación de la asignatura, descripción del programa, normativa y los métodos de evaluación, descripción de la plataforma Moodle. - Tema 1. Punteros y memoria dinámica en C. Apartado 1.1 - Práctica 1	5	3 Trabajo del estudiante: Lectura de las normativas de teoría y prácticas. Familiarización con el entorno Moodle. Lectura del material propuesto sobre el tema 1. Realización Práctica 1.
2	- Tema 1. Punteros y memoria dinámica en C. Apartado 1.2	5	3 Trabajo del estudiante: Lectura del material



Asignatura: Programación II
 Código: 18468
 Centro: Escuela Politécnica Superior
 Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
 Nivel: Grado
 Tipo: Formación básica
 Nº de créditos: 6

Semana	Contenido	Horas presenciales	Horas no presenciales
	- Práctica 1		propuesto sobre el tema 1. Realización Práctica 1.
3	- Tema 2 Introducción - Tema 3 Listas. Apartados 3.1 y 3.2 - Práctica 2	5	2 Trabajo del estudiante: Lectura del material propuesto sobre los temas 2 y 3. Realización de Práctica 2.
4	- Tema 3 Listas. Apartados 3.3, 3.4, 3.5 y 3.6 - Práctica 2	5	3 Trabajo del estudiante: Lectura de material propuesto sobre el tema 3. Realización de los ejercicios propuestos. Realización de Práctica 2.
4	- Tutorías sobre los temas 2 y 3	1	
5	- Tema 4 Pilas - Tema 5 Colas. Apartados 5.1, 5.2 y 5.3 - Práctica 2	5	2 Trabajo del estudiante: Lectura de material propuesto sobre los temas 4 y 5. Realización de los ejercicios propuestos. Entrega de Práctica 2.
5	- Tutorías sobre los temas 4 y 5	1	
6	- Tema 5 Colas. Apartados 5.4, 5.5, 5.6 y 5.7 - Primera prueba intermedia - Práctica 3	5	3 Trabajo del estudiante: Lectura de material propuesto sobre el tema 5. Realización de los ejercicios propuestos. Realización de Práctica 3.
7	- Tema 6 Árboles. Apartados 6.1, 6.2, 6.3 y 6.4 - Práctica 3	5	2 Trabajo del estudiante: Lectura de material propuesto sobre el tema 6. Realización de los ejercicios propuestos. Realización de Práctica 3.
8	- Tema 6 Árboles. Apartados 6.5 y 6.6 - Práctica 3	5	3 Trabajo del estudiante: Lectura de material propuesto sobre el tema 6. Realización de los ejercicios propuestos. Realización de Práctica 3.
8	- Tutorías sobre el tema 6	1	
9	- Tema 7 Grafos. Apartados 7.1, 7.2 y 7.3 - Práctica 3	5	3 Trabajo del estudiante: Lectura de material propuesto sobre el tema



Asignatura: Programación II
 Código: 18468
 Centro: Escuela Politécnica Superior
 Titulación: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
 Nivel: Grado
 Tipo: Formación básica
 Nº de créditos: 6

Semana	Contenido	Horas presenciales	Horas no presenciales
			7. Realización de los ejercicios propuestos. Entrega de Práctica 3.
10	- Tema 7 Grafos. Apartados 7.4 y 7.5 - Prueba intermedia 2 - Práctica 4	5	2 Trabajo del estudiante: Lectura de material propuesto sobre el tema 7. Realización de los ejercicios propuestos. Realización de Práctica 4.
10	- Tutorías sobre el tema 7	1	
11	- Tema 8. Algoritmos de ordenación. Apartados 8.1, 8.2, 8.3, 8.4 y 8.5 - Práctica 4	5	3 Trabajo del estudiante: Lectura de material propuesto sobre el tema 8. Realización de los ejercicios propuestos. Realización de Práctica 4.
12	- Tema 8. Algoritmos de ordenación. Apartado 8.6 - Tema 9. Algoritmos de búsqueda. Apartados 9.1 y 9.2 - Práctica 4	5	3 Trabajo del estudiante: Lectura de material propuesto sobre los temas 8 y 9. Realización de los ejercicios propuestos. Realización de Práctica 4.
13	- Tema 9. Algoritmos de búsqueda. Apartado 9.3 - Práctica 4	5	3 Trabajo del estudiante: Lectura de material propuesto sobre el tema 9. Realización de los ejercicios propuestos. Entrega de Práctica 4.
14	- Tema 10. Análisis de complejidad	5	2 Trabajo del estudiante: Lectura de material propuesto sobre el tema 10. Realización de los ejercicios propuestos.
14	- Tutorías sobre los temas 8, 9 y 10	1	
	Examen Final	6	32