



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

## GUÍA DOCENTE DE AMPLIACIÓN DE PROGRAMACIÓN

La presente guía docente corresponde a la asignatura Ampliación de Programación (AMP-PROG), 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.



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

## ASIGNATURA

### AMPLIACIÓN DE PROGRAMACIÓN (AMP-PROG)

#### 1.1. Código

18767 del Grado en Ingeniería Informática

#### 1.2. Materia

Ingeniería del software y Sistemas de Información

#### 1.3. Tipo

Optativa

#### 1.4. Nivel

Grado

#### 1.5. Curso

4º

#### 1.6. Semestre

1º

#### 1.7. Número de créditos

6 créditos ECTS

#### 1.8. Requisitos previos

Esta asignatura se imparte en el primer semestre del cuarto curso. AMP-PROG forma parte de las *Materias 1* y *4* (“Ingeniería del Software”, 18 ECTS, “Sistemas de Información”, 24 ECTS) de las asignaturas optativas de cuarto curso. El único requisito académico específico es tener conocimientos de programación en lenguajes como C o Java.



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

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 ([moodle.uam.es](http://moodle.uam.es)) y la búsqueda activa de material complementario en la red. Es recomendable disponer de un dominio de inglés que permita al alumno leer la bibliografía de consulta. Asimismo, se requiere iniciativa personal y constancia para el diseño/escritura de programas y la resolución de ejercicios durante el curso. Finalmente, se requiere 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 evaluación continua supondrá una asistencia de al menos un 80% de las horas.

## 1.10. Datos del equipo docente

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

### Profesores de teoría:

**Dr. Francisco Saiz López (Coordinador)**

Departamento de Ingeniería Informática

Escuela Politécnica Superior

Despacho - Módulo: B-323 Edificio B - 3ª Planta

Teléfono: +34 91 497 2281

Correo electrónico: francisco.saiz

Página web: <http://www.ii.uam.es/~saiz>

Horario de atención al alumnado: Petición de cita previa por correo electrónico.



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

#### Profesores de prácticas:

**Francisco Saiz López** (Coordinador de prácticas)

Departamento de Ingeniería Informática

Escuela Politécnica Superior

Despacho - Módulo: B-323 Edificio B - 3ª Planta

Teléfono: +34 91 497 2281

Correo electrónico: francisco.saiz

Página web: <http://www.ii.uam.es/~saiz>

Horario de atención al alumnado: Petición de cita previa por correo electrónico.

## 1.11. Objetivos del curso

AMP-PROG es una asignatura de ampliación de las técnicas de programación tratadas en la titulación, que en su mayoría han sido relativas a lenguajes imperativos como Java o C. Se estudiará inicialmente la **programación funcional**, con el lenguaje **Haskell** como referencia académica. Además, incluirá aplicaciones en el entorno de **desarrollo .NET**, usando el lenguaje de programación **C#** (orientado a objetos con muchas componentes de programación funcional). El curso también será, por tanto, una ampliación de los conceptos de POO del lenguaje Java del segundo curso de la titulación, en el contexto de .NET, incorporando el uso de un paradigma de programación diferente al imperativo, que resulta ser especialmente **conciso, elegante y robusto**.

Las competencias que se pretenden adquirir con esta asignatura son:

- **C6. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.**
- **C7. Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.**
- **C8. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados**
- **IS3. Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.**
- **SI1. Capacidad de integrar soluciones de Tecnologías de la Información y las Comunicaciones y procesos empresariales para satisfacer las necesidades de información de las organizaciones, permitiéndoles alcanzar sus objetivos de forma efectiva y eficiente, dándoles así ventajas competitivas.**
- **CC4. Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir**



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

**sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.**

Los resultados del aprendizaje que el estudiante adquiere con la asignatura AMP-PROG son:

- Capacidad para resolver problemas utilizando enfoques diferenciados en su expresión como lenguajes de programación.
- Conocimiento de nuevos entornos integrados con lenguajes de propósito general con componentes mixtas.
- Capacidad para determinar la idoneidad de los lenguajes usados para resolver problemas.
- Capacidad para determinar los métodos para abordar problemas complejos, en función de la forma en que se resuelven y se expresan en un determinado lenguaje.

OBJETIVOS ESPECIFICOS POR TEMA	
<b>TEMA 1.- Introducción a la programación funcional</b>	
1.1.	Revisar diferentes enfoques o paradigmas en programación
1.2.	Presentar aspectos característicos de la programación funcional
<b>TEMA 2.- Programación funcional en un lenguaje puro: Haskell</b>	
2.1.	Introducir sus principales características
2.2.	Incorporar dentro del lenguaje a los tipos
2.3.	Estudiar en detalle el tratamiento de funciones
2.4.	Analizar el tratamiento de operadores dentro del lenguaje
2.5.	Estudiar las clases de tipos
2.6.	Hacer un recorrido por los módulos o bibliotecas más importantes
2.7.	Introducir los conceptos de mónadas y su aplicación para entrada-salida
2.8.	Presentar ejemplos o aplicaciones más elaboradas sobre el lenguaje
<b>TEMA 3.- Desarrollo en .NET (lenguaje C#)</b>	
3.1.	Presentar tipos y, en particular, los más importantes: clases y estructuras
3.2.	Analizar el método de acceso a elementos mediante propiedades e indizadores
3.3.	Introducir las componentes más importantes de tipo colección
3.4.	Presentar los conceptos de delegados y eventos
3.5.	Estudiar ejemplos de aplicaciones sencillas en C# usando Windows Forms
3.6.	Analizar las operaciones de polimorfismo y tratamiento de funciones anónimas en C#
3.7.	Estudiar el uso de genéricos en C#
3.8.	Estudiar el uso de métodos de extensión en C#
3.9.	Estudiar la paralelización de programas con TPL
3.10.	Presentar una iniciación al lenguaje F#
<b>TEMA 5.- Temas avanzados</b>	
5.1.	Estudiar conceptos de patrones de diseño clásicos y funcionales
5.2.	Presentar la idoneidad en términos de uso en concurrencia o en el caso MapReduce



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

## 1.12. Contenidos del programa

### Programa Sintético

UNIDAD 1. Introducción a la programación funcional  
UNIDAD 2. Programación funcional en un lenguaje puro: Haskell  
UNIDAD 3. Desarrollo en .NET (lenguaje C#)  
UNIDAD 5. Temas avanzados

### Programa Detallado

1. **Introducción a la programación funcional**
  - 1.1. Paradigmas de programación
  - 1.2. Ventajas e inconvenientes de uso
2. **Programación funcional en un lenguaje puro: Haskell**
  - 2.1. Introducción
  - 2.2. Tipos
  - 2.3. Funciones
  - 2.4. Operadores
  - 2.5. Clases
  - 2.6. Bibliotecas
  - 2.7. Entrada-salida
  - 2.8. Aplicaciones
3. **Desarrollo en .NET (lenguaje C#)**
  - 3.1. Tipos, clases y estructuras
  - 3.2. Propiedades e indizadores
  - 3.3. Colecciones
  - 3.4. Delegados y eventos
  - 3.5. Windows Forms
  - 3.6. Polimorfismo y funciones anónimas
  - 3.7. Elementos genéricos
  - 3.8. Métodos de extensión
  - 3.9. Biblioteca TPL
  - 3.10. Introducción a F#
4. **Temas avanzados**
  - 4.1. Patrones clásicos y funcionales
  - 4.2. Concurrencia, MapReduce



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

## 1.13. Referencias de consulta

[Bibliografía disponible a través del catálogo de la biblioteca \(pincha aquí\)](#)

### Bibliografía:

1. <http://www.haskell.org> (“books & tutorials”)
2. “Haskell the craft of functional programming”, Thompson, S. Ed. Addison Wesley, 2011.
3. “Programming in Haskell”, Hutton, G. Ed. Cambridge University Press, 2007.
4. <http://msdn.microsoft.com/en-us/vstudio> (“master the languages”)
5. “Essential C# 4.0”, Michaelis, M. Ed. Addison-Wesley, 2010.
6. “Functional programming in C# classic programming techniques for modern projects”, Sturm, O. Ed. John Wiley, 2011.
7. “Microsoft Visual C# 2010 step by step”, Sharp, J. Ed. Microsoft, 2010.
8. “Programming F#”. Smith, C. Ed. O’Reilly, 2009.
9. “Patrones de diseño para C# Los 23 modelos de diseño: descripción y soluciones ilustradas en UML2 y C#”, Debrauwer, L. Ed. ENI, 2012.

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 AMP-PROG en plataforma Moodle ([moodle.uam.es](http://moodle.uam.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

Clases expositivas simultaneadas con la realización de programas y ejercicios. Se utilizará la pizarra, combinada con la explicación de programas en formato electrónico cuya ejecución se visualizará en la pantalla de la clase.

#### Actividad del estudiante:

*Actividad presencial:* Toma de apuntes, participación activa en clase, resolución de algunos ejercicios propuestos.

*Actividad no presencial:* Lectura del material bibliográfico y de apoyo, estudio de la materia y realizaciones de ejercicios propuestos.



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

**\*Clases de problemas/ejercicios en aula/laboratorio:**

**Actividad del profesor**

Primera parte expositiva, una segunda parte de supervisión y asesoramiento en la resolución de los problemas y una parte final de análisis del resultado y generalización.

**Actividad del estudiante:**

*Actividad presencial:* Participación activa en la resolución de los ejercicios, diseño y escritura de los programas y en el análisis de la ejecución.

*Actividad no presencial:* Realización de ejercicios y programas, planteados en clase. Generalización y planteamiento de modificaciones que permitan la optimización de los programas.

**\*Tutorías:**

**Actividad del profesor:**

Tutorías con el objetivo de resolver dudas comunes plantadas por los alumnos a nivel individual o en grupo, surgidas a partir de cuestiones, ejercicios o 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.

**\*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.

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. Al finalizar la práctica se entrega un breve informe con el programa desarrollado y, además, se podrá solicitar la ejecución con el profesor presente, quien podrá hacer las preguntas oportunas a cada miembro del grupo para calificar de forma individual la práctica.



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

### 3. Tiempo de trabajo del estudiante

		Nº de horas	Porcentaje
Presencial	Clases teóricas	42 h (28%)	72 h (48%)
	Clases prácticas	24 h (16%)	
	Realización de prueba final	6 h (4%)	
No presencial	Estudio semanal	46 h (31%)	78 h (52%)
	Preparación de pruebas	32 h (21%)	
<b>Carga total de horas de trabajo: 25 horas x 6 ECTS</b>		<b>150 h</b>	

### 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:

$$\text{Calificación: } 0.3 \cdot \text{Prácticas} + 0.7 \cdot \text{Teoría}$$

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

$$\text{Calificación: } \text{Mín}(4.9, 0.3 \cdot \text{Prácticas} + 0.7 \cdot \text{Teoría})$$

- La nota correspondiente a la parte de Teoría para el método de evaluación continua es la que resulta de obtener el máximo entre "A" y "B":

- A) La calificación de la prueba final.
- B) La calificación de las pruebas intermedias. La calificación de estas pruebas intermedias se obtendrá aplicando la siguiente fórmula:

$$\text{Calificación\_intermedias} = 0,25 \cdot p1 + 0,35 \cdot p2 + 0,4 \cdot p3$$

- Observaciones:

- La tercera prueba intermedia coincidirá en fecha con la prueba final.
- El alumno deberá elegir entre realizar la tercera prueba intermedia o la prueba final. Aclaración: si realiza la prueba final, se considerará  $p3=0$  en la fórmula anterior.



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

➤ 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:

$$\text{Calificación\_prácticas} = p1 * e1 + p2 * e2 + p3 * e3$$

donde e1, e2 y e3 son las calificaciones obtenidas en las entregas intermedias y p1, p2 y p3 ( $p1+p2+p3=1$ ) son, en términos relativos, la cantidad de sesiones docentes dedicadas en el laboratorio a cada entrega (idealmente  $p1=p2=p3=1/3$ ).

La nota final será diferente a “No evaluado” si y sólo si se ha realizado la prueba final, o dos de las pruebas intermedias, o 2 de las 3 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.
  - Una nota de prácticas de la convocatoria ordinaria se puede conservar (convalidar) sólo para la convocatoria extraordinaria del mismo curso.

**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.  - <b>Unidad 1</b> Introducción a la PF. Temas 1.1 y 1.2	3	5 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados.
2	- <b>Unidad 2.</b> Haskell. Temas 2.1 y 2.2  - Práctica 1	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales



Asignatura: Ampliación de Programación  
 Código: 18767  
 Centro: Escuela Politécnica Superior  
 Titulación: Grado en Ingeniería Informática  
 Nivel: Grado  
 Tipo: Optativa  
 Nº de créditos: 6

Semana	Contenido	Horas presenciales	Horas no presenciales
			tratados. Realización de la Práctica 1.
3	- Unidad 2. Haskell. Tema 2.3 y 2.4 - Práctica 1	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 1.
4	- Unidad 2. Haskell. Tema 2.5 - Práctica 1	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 1.
5	- Unidad 2. Haskell. Tema 2.6 - Práctica 1 - Primera prueba intermedia	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 1.
6	- Unidad 2. Haskell. Tema 2.7 - Práctica 2	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 1.
7	- Unidad 2. Haskell. Tema 2.8 - Práctica 2	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 1.
8	- Unidad 3. .NET. Temas 3.1 y 3.2 - Práctica 2	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Entrega de la Práctica 1.
9	- Unidad 3. .NET. Temas 3.3 y 3.4 - Práctica 2 - Segunda prueba intermedia	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 2.



Asignatura: Ampliación de Programación  
Código: 18767  
Centro: Escuela Politécnica Superior  
Titulación: Grado en Ingeniería Informática  
Nivel: Grado  
Tipo: Optativa  
Nº de créditos: 6

Semana	Contenido	Horas presenciales	Horas no presenciales
10	- Unidad 3. .NET. Tema 3.5 - Práctica 3	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 2.
11	- Unidad 3. .NET. Temas 3.6 y 3.7 - Práctica 3	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 2.
12	- Unidad 3. .NET. Temas 3.8 y 3.9 - Práctica 3	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Realización de la Práctica 2.
13	- Unidad 3. .NET. Tema 3.10 - Práctica 3	5	3 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados. Entrega de la Práctica 2.
14	- Unidad 4. Temas avanzados. Temas 4.1, 4.2	3	5 Trabajo del estudiante: realización de la asignación propuesta sobre los temas semanales tratados.
	Examen Final	6	32