Asignatura: Proyecto de Programación
Código: 17823
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática /
Doble Grado en Ingeniería Informática y en Matemáticas
Nivel: Grado
Tipo: Formación obligatoria
Nº de créditos: 6

GUÍA DOCENTE DE PROYECTO DE PROGRAMACIÓN

La presente guía docente corresponde a la asignatura Proyecto de Programación (PPROG), aprobada para el curso lectivo 2016-2017 por la Junta de Centro de la Escuela Politécnica Superior y publicada en su versión definitiva en la página web de la misma escuela. Esta guía docente de PPROG es aprobada y publicada antes del periodo de matrícula.

Asignatura

PROYECTO DE PROGRAMACIÓN (PPROG)

1.1. Código

17823 de la titulación Grado en Ingeniería Informática y de la titulación Doble Grado en Ingeniería Informática y en Matemáticas.

1.2. Materia

Programación

1.3. Tipo

Formación obligatoria

1.4. Nivel

Grado

1.5. Curso

1º de la titulación Grado en Ingeniería Informática.
2º de la titulación Doble Grado en Ingeniería Informática y en Matemáticas.

1.6. Semestre

2º de la titulación Grado en Ingeniería Informática.
1º de la titulación Doble Grado en Ingeniería Informática y en Matemáticas.

1.7. Número de créditos

6 créditos ECTS

1.8. Requisitos previos

Es requisito para la correcta adquisición de los contenidos, habilidades y destrezas de la asignatura tener buena capacidad de: lectura comprensiva de textos en castellano e inglés, búsqueda e identificación de bibliografía y redacción de memorias en lengua castellana o inglesa.

Es altamente recomendable haber cursado y superado las asignaturas **Programación I** y **Seminario-Taller de Software**. En la primera de ellas se adquieren los conocimientos y destrezas de programación necesarios para cursar esta asignatura. En Seminario-Taller de Software se obtienen las capacidades para desenvolverse en el sistema operativo de trabajo, así como manejar entornos y herramientas para desarrollo, gestión de proyectos y trabajo colaborativo que se utilizan en el Proyecto de Programación (PPROG).

PPROG pertenece al módulo de "Programación y Estructuras de Datos," encuadrándose dentro de la materia de "Programación" junto a las asignaturas *Programación I* y *Programación II*. PPROG complementa los resultados de aprendizaje, competencias y contenidos de las asignaturas de dicho módulo.

PPROG sirve de base para los conocimientos, habilidades y aptitudes que se adquirirán en las asignaturas *Proyecto de Análisis y Diseño de Software*, *Proyecto de Autómatas y Lenguajes*, *Proyecto de Sistemas Informáticos* y *Proyecto de Ingeniería del Software*.

1.9. Requisitos mínimos de asistencia a las sesiones presenciales

Se plantean dos itinerarios: uno con asistencia obligatoria a clase y otro sin ella. Los estudiantes deberán optar por uno u otro a principio del curso y cumplir con los distintos requisitos de evaluación que conlleva cada uno de ellos, según se especifican en la presente guía docente (véase el apartado 4).

ITINERARIO CON EVALUACIÓN CONTINUA Y ASISTENCIA OBLIGATORIA A CLASE
En este itinerario es obligatorio asistir al 85% de las sesiones presenciales.

ITINERARIO CON EVALUACIÓN FINAL Y SIN ASISTENCIA OBLIGATORIA A CLASE
En este itinerario no se exige asistir a todas las sesiones presenciales, aunque se considera muy recomendable, pero sí presentar los entregables que se establezcan en las fechas indicadas para ello a principio de curso (véase el apartado 4).

Los estudiantes podrán cambiar el itinerario, previa autorización del profesor o por indicación de este; el profesor podrá poner un límite de tiempo al cambio voluntario de itinerario. Este límite, si existe, será comunicado a los estudiantes los primeros días de clase.

Todos los estudiantes deberán estar presentes en las sesiones de evaluación establecidas como tales para sus itinerarios.

1.10. Datos del equipo docente

En las direcciones de correo electrónico hay que añadir @uam.es

Dr. Alejandro Bellogín Kouki
Departamento de: Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-434
Teléfono: +34 91 497 2256
Correo electrónico: alejandro.bellogin
Página web: http://www.eps.uam.es/~abellogin
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Ruth Cobos Pérez
Departamento de: Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-427
Teléfono: +34 91 497 2243
Correo electrónico: ruth.cobos
Página web: http://www.eps.uam.es/~rcobos
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Fernando Díez Rubio
Departamento de: Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-409
Teléfono: +34 91 497 2213
Correo electrónico: fernando.diez
Página web: http://www.eps.uam.es/~fdiez
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. David Domínguez Carrera
Departamento de: Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-425
Teléfono: +34 91 497 2245
Correo electrónico: david.dominguez
Página web: http://www.eps.uam.es/~ddominguez
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Javier Gómez Escribano
Departamento de: Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-403
Teléfono: +34 91 497 7541
Correo electrónico: jg.escribano
Página web: https://javiergomezescribano.wordpress.com/
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Francisco Jurado Monroy (coordinador)
Departamento de: Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-344
Teléfono: +34 91 497 7527
Correo electrónico: francisco.jurado
Página web: http://www.eps.uam.es/~fjurado
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Jaime Moreno Llorena
Departamento de: Ingeniería Informática
Escuela Politécnica Superior
Despacho: B-347
Teléfono: +34 91 497 2212
Correo electrónico: jaime.moreno
Página web: http://www.eps.uam.es/~jmoreno
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Simone Santini (coordinador Doble Grado en Informática y en Matemáticas)
Departamento de: Ingeniería informática
Escuela Politécnica Superior
Despacho: B-350
Teléfono: +34 91 497 2209
Correo electrónico: simone.santini
Página web: http://www.eps.uam.es/~ssantini
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

1.11. Objetivos del curso

Este curso forma parte del módulo de “Programación y Estructuras de Datos” y corresponde a la materia de “Programación”. Su objetivo principal es proporcionar al alumno las capacidades para crear una aplicación compleja trabajando como parte de un equipo. Para ello, el alumno tendrá que emplear conocimientos y capacidades que se adquieren en el primer curso de los grados de los que la asignatura forma parte, antes o durante el desarrollo de la misma.

Este objetivo se encuentra íntimamente relacionado con la siguiente competencia del grado:

9. Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión correspondiente al grado en Ingeniería Informática.

Las **competencias** comunes a la rama informática que PPROG contribuye a alcanzar son:

B4. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos, y programas informáticos con aplicación en ingeniería.

C3. Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.

C4. Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.

C5. Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas.

Los **objetivos generales** que se pretenden alcanzar con esta asignatura son:

G1. Trabajar en equipo de manera efectiva para la consecución de un programa.

G2. Redactar la documentación de un programa informático.

G3. Diseñar, implementar e integrar los diferentes módulos de un programa.

G4. Verificar y validar un programa.

Los **objetivos específicos** para cada unidad didáctica son:

Unidad 1. Trabajo en Equipo

1.1) Adquirir compromisos y responsabilidades frente a un equipo de trabajo.
1.2) Elaborar una planificación razonable en tiempo y recursos.
1.3) Coordinarse para la consecución de un objetivo común.
1.4) Integrar código de diferentes autores.

Unidad 2. Codificación y documentación

2.1) Codificar siguiendo una guía de estilo.
2.2) Documentar el código.
2.3) Redactar documentación técnica.

Unidad 3. Pruebas

3.1) Diseñar pruebas.
3.2) Redactar un documento de plan de pruebas.
3.3) Aplicar el plan de pruebas a un programa.

Unidad 4. Módulos y bibliotecas

4.1) Trabajar en un programa de manera modular.
4.2) Buscar e identificar bibliotecas existentes.
4.3) Usar bibliotecas de terceros.

Unidad 5. Proyecto

5.1) Implementar un proyecto de programación en base a un documento de requisitos y un diseño preliminar parcial.
5.2) Integrar los conceptos y capacidades adquiridas en las unidades anteriores obteniendo como resultado un programa informático complejo integrado por diversos módulos.

1.12. Contenidos del programa

El contenido de esta asignatura se articula en torno a un proyecto de programación que de forma general se implementará en el lenguaje de programación ANSI C y deberá poderse compilar, enlazar y depurar en un terminal utilizando herramientas de desarrollo GNU (make, gcc y gdb). El entorno de desarrollo a utilizar es de libre elección por parte del estudiante, pero el proyecto entregado deberá contar con un Makefile escrito directamente por los alumnos para la construcción de la aplicación desde la línea de comandos de un terminal con make y gcc de GNU.

La implementación del proyecto parte de una especificación técnica y de una propuesta de diseño modular básico. La concreción de los diferentes módulos del proyecto se aproximará en las distintas unidades de la asignatura en consonancia con los objetivos asociadas a cada una de ellas. A continuación se detalla el programa de la asignatura.

Programa Detallado

1. Introducción a la asignatura
   1.1. Normativa y temario
   1.2. Presentación del proyecto: requisitos y aproximación iterativa
2. Codificación y documentación
   2.1. Uso del entorno de desarrollo GNU: make, gcc y gdb
   2.2. Guías de estilo de codificación
   2.3. Documentación técnica con Doxygen
   2.4. Conceptos de programación modular: módulos y bibliotecas
3. Introducción a la verificación y validación
   3.1. Trazas de ejecución
   3.2. Pruebas software
4. Realización y gestión de un proyecto
   4.1. Introducción a la gestión y planificación de proyectos
   4.2. Realización de un proyecto de forma iterativa e incremental siguiendo fases de diseño, implementación, depuración, integración y pruebas

1.13. Referencias de consulta

1. Kernighan, Brian W y Pike, Rob. **The Practice of Programming**. Addison-Wesley. ISBN: 020161586X. 1999. Cat: INF/C6110/KER (Texto completo en línea) (Disponible también en castellano)
2. Kernighan, Brian W. y Ritchie, Dennis M. **The C programming language**. Prentice Hall. ISBN: 0131103628. 1988. Cat: INF/C6140-C/KER (Texto completo en línea) (Disponible también en castellano)

Nota: se recomienda a los estudiantes no comprar libro alguno hasta haberlo revisado previamente en la biblioteca.

Material electrónico de trabajo: los documentos electrónicos adicionales se publicarán en las secciones de PPROG establecidas cada curso en la plataforma Moodle (https://moodle.uam.es/).

2. Métodos docentes

Para alcanzar los objetivos previstos (véase apartado 1.11), la asignatura se plantea como un proceso de desarrollo práctico iterativo por equipos, en el que mientras se realizan aproximaciones sucesivas a la aplicación final, se van desarrollando las habilidades e introduciendo los conocimientos necesarios de forma progresiva. Cada una de las iteraciones se concreta en la implementación de un programa operativo que incorpora parte de la funcionalidad de la aplicación final, de forma que cada aproximación se basa en el resultado de la iteración anterior.

Para ello la asignatura se organiza como un conjunto de sesiones prácticas presenciales complementadas con trabajo no presencial estrechamente relacionado con ellas. El temario se va dosificando según se requiere a lo largo del curso y del proceso de desarrollo iterativo. La metodología utilizada se fundamenta en aprendizaje basado en proyectos y en el aprendizaje autónomo del alumno. En el desarrollo de la actividad docente se incluyen los siguientes tipos de actividades:

* **Sesiones prácticas de laboratorio**

Actividad del profesor:

Al comienzo de cada semana el profesor presentará los objetivos y organizará las actividades que se vayan a llevar a cabo en las sesiones presenciales y aquellas que deberían realizarse en otro momento a lo largo del periodo. Durante estas sesiones el profesor explicará y suministrará la documentación para las actividades programadas, supervisará el trabajo de los equipos y ayudará a resolver las dudas y problemas que puedan surgir.

Se contemplan distintos tipos de actividades a realizar en las sesiones presenciales:

a. Propuesta, análisis y desarrollo de casos prácticos.
b. Técnicas de trabajo en equipo y tecnologías para la colaboración.
c. Presentación, revisión y comprensión del material docente.

Los medios utilizados son el hardware y el software disponibles en los laboratorios (o infraestructura equivalente) para el desarrollo, ejecución y prueba de los programas realizados, así como la plataforma docente institucional (Moodle) y el material didáctico proporcionado a través de ella.

Actividad del estudiante:

Actividad presencial: Depende de la actividad planteada en cada sesión se elaboran los materiales requeridos para los entregables programados.

Actividad no presencial: Profundizar en el enunciado de las actividades propuestas. Al finalizar los periodos establecidos se presentan los correspondientes entregables, que se evalúan en sesiones presenciales específicas.

*** <u>Tutorías en aula</u>**

Actividad del profesor:

Las tutorías se organizan bien para (1) realizar un seguimiento de los grupos más personalizado y resolver conflictos entre los integrantes; bien para (2) resolver dudas comunes planteadas por los alumnos a nivel individual o en grupo, surgidas a partir de las prácticas o el proyecto.

Actividad del estudiante:

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

*** <u>Reuniones de equipo</u>**

Actividad del estudiante:

Actividad no presencial consistente en las reuniones cara a cara o en línea entre los miembros del equipo que estos consideren necesarias para tratar temas de coordinación, integración de los resultados parciales realizados por cada uno de ellos, etc.

*** <u>Lecturas obligatorias, estudio y trabajo personal</u>**

Actividad del estudiante:

Actividad no presencial consistente en el aprendizaje autónomo académicamente dirigido por el profesor a través de las tareas publicadas en la página de docencia en red y tareas de análisis, diseño y codificación.

3. Tiempo de trabajo del estudiante

		Nº de horas	Porcentaje
Presencial	Clases teóricas	0 h (0%)	54 h (36%)
	Clases prácticas	48 h (32%)	
	Realización de la prueba final	6 h (4%)	
No presencial	Realización de actividades prácticas en laboratorio (**2.5h x 14 semanas**)	35 h (23%)	96 h (64%)
	Trabajo y estudio individual regulado	30 h (20%)	
	Trabajo y estudio individual **no** regulado para la preparación de las convocatorias ordinaria y extraordinaria	31 h (21%)	
Carga total de horas de trabajo: 25 horas x 6 ECTS		**150 h**	

Tabla 1. Repartición del tiempo de trabajo del estudiante

4. Métodos de evaluación y porcentaje en la calificación final

La evaluación de la asignatura estará asociada a **entregables**, con posibles entregas intermedias vinculadas a ellos que se considerarán en su evaluación, así como a una **prueba final de evaluación individualizada**. Cada entregable es el resultado del trabajo en equipo de un grupo de alumnos, de forma que el objetivo de la prueba individual es valorar el grado de participación de cada uno de los miembros de un equipo de trabajo en los entregables, así como del aprovechamiento individual conseguido. La Tabla 2 muestra el cálculo de la nota final de la asignatura para los dos itinerarios establecidos.

NOTA FINAL (NF) **Itinerario CON asistencia obligatoria a clase**	**NF: 0,8*NE + 0,2*NPI** donde NPI es la nota de la prueba individual y NE representa una ponderación de las calificaciones correspondientes a los entregables asociados a las iteraciones del proyecto según se especificará al principio del curso. Para aprobar la asignatura tiene que cumplirse que: a) NF >= 5 puntos b) NE >= 5 puntos c) NPI >= 4,5 puntos d) La nota de cada entregable >= 4,5 puntos (Los entregables no presentados y las pruebas no realizadas tendrán una calificación de 0 puntos.) En caso de no alcanzarse dichas calificaciones, la nota final en actas será: **NF: 0,8*Mín(4,5; NE) + 0,2*Mín(4,5; NPI)** Si la NF es 0 la calificación final será No Evaluable.
NOTA FINAL (NF) **Itinerario SIN asistencia obligatoria a clase**	**NF: 0,5*NE + 0,5*NPI** donde NPI es la nota de la prueba individual y NE representa una ponderación de las calificaciones correspondientes a los entregables asociados a las iteraciones del proyecto según se especificará al principio del curso. Para aprobar la asignatura tiene que cumplirse que: a) NF >= 5 puntos b) NE >= 5 puntos c) NPI >= 5 puntos d) La nota de cada entregable >= 4,5 puntos (Los entregables no presentados y las pruebas no realizadas tendrán una calificación de 0 puntos.) En caso de no alcanzarse dichas calificaciones, la nota final en actas será: **NF: 0,5*Mín(4,5; NE) + 0,5*Mín(4,5; NPI)** Si la NF es 0 la calificación final será No Evaluable.

Tabla 2. Sistema de Evaluación de la Asignatura Proyecto de Programación

Consideraciones generales

1. Es obligatorio que cada alumno esté presente en las **sesiones de evaluación** establecidas para que la parte correspondiente compute, tanto si se sigue el itinerario con asistencia obligatoria como sin ella.
2. Cualquier **copia** descubierta que se haya realizado a lo largo del curso será penalizada por igual, tanto a los alumnos que copian como a los copiados. La penalización por copia implica la aplicación de la normativa interna de la EPS.

Entregables

3. Los entregables se evaluarán siguiendo una **rúbrica** de corrección común en todos los grupos que será publicada junto con los enunciados.
4. Los entregables se califican de **0 a 10 puntos** y se consideraran aceptables si su calificación es **igual o superior a 4,5 puntos**.
5. Para optar a superar la asignatura es imprescindible **haber presentado en los periodos establecidos todos los entregables** según la planificación determinada al principio del curso para cada itinerario, y que **todos ellos sean aceptables**.
6. La evaluación de cada entregable puede requerir la realización de una **entrevista** por equipo donde se evaluará el desarrollo del mismo, así como los conocimientos de la asignatura que han empleado para ello. En la evaluación de la última iteración se podría solicitar la presentación del proyecto final a toda la clase.

Proyecto

7. El enunciado donde se incluye un diseño de alto nivel del proyecto estará publicado en Moodle desde principio de curso. Este enunciado será único para todos los equipos.

Prueba individual final

8. La prueba individual final en cada convocatoria se realizará sólo si se **ha conseguido una nota mayor o igual a 5 puntos en el cálculo de NE** (ver Tabla 2)**, siendo todos los entregables aportados aceptables**.
9. La prueba individual final consistirá en un **examen escrito** de duración estimada de **3 horas**.

Composición de equipos

10. Las actividades se realizarán en **equipos que podrán incrementar su tamaño desde las primeras iteraciones a las últimas**.
11. La composición de los equipos se establecerá al inicio de cada iteración entre alumnos en el mismo itinerario, **según los criterios que se establezcan al efecto a principio del curso**.
12. En caso de problemas en los equipos, la **restructuración** sólo se podrá realizar con autorización del profesor o por indicación de este y bajo su supervisión. Si en la restructuración no se consiguen nuevos integrantes para completar el grupo de trabajo, este se disolverá y los miembros se reasignarán a otros equipos.
13. La calificación de un **alumno reasignado** a otro grupo se obtendrá utilizando las notas del material entregado con el grupo anterior hasta el momento de la reasignación y las notas del material desarrollado junto con el nuevo grupo. El alumno podrá reutilizar en el nuevo equipo todo aquel código del que se le haya atribuido la autoría.

Convocatoria extraordinaria

14. Para poder optar a la evaluación extraordinaria en ambos itinerarios, es necesario **haber presentado todos los entregables** requeridos antes de la fecha límite que se indique y **haber conseguido una nota mayor o igual a 5 puntos en el cálculo de NE** (ver Tabla 2)**, siendo todos los entregables aceptables**.
15. Aquellos entregables que no se hubieran podido evaluar antes, se evaluarán la semana antes de la prueba individual final de la convocatoria extraordinaria.

5. Cronograma

El cronograma presentado a continuación y algunos contenidos a los que hace referencia podrían ajustarse a lo largo del curso. Además, no se incluyen las fechas concretas de los entregables. Estas fechas se determinarán al comienzo del curso y se comunicarán convenientemente, igual que los posibles ajustes, tanto en clase como a través de Moodle.

Semana	Contenido	Horas presenciales	Horas no presenciales
1	Presentación de la asignatura y del proyecto a desarrollar iterativamente. Comienzo de trabajo en el proyecto.	3	5
2	Revisión del entorno básico de desarrollo GNU (gcc, make y gdb). Trabajo en proyecto y seguimiento del mismo.	3	5
3	Guías de estilo de programación y documentación técnica del código (Doxygen). Trabajo en proyecto y seguimiento del mismo.	3	5
4	Fundamentos sobre trabajo en grupo. Trabajo en proyecto y seguimiento del mismo.	3	5
5	Módulos y bibliotecas. Trabajo en proyecto y seguimiento del mismo.	3	5
6	Trazas de programas. Trabajo en proyecto y seguimiento del mismo.	3	5
7	Introducción a la gestión y planificación de proyectos. Trabajo en proyecto y seguimiento del mismo.	3	5
8	Introducción a las pruebas de software. Trabajo en proyecto y seguimiento del mismo.	3	5
9-14	Trabajo en proyecto y seguimiento del mismo.	18	30
15	Revisión final del proyecto realizado.	3	5
	Prueba individual final ordinaria.	3	12
	Prueba individual final extraordinaria.	3	12