

**Proyecto/Guía docente de la asignatura**

<b>Asignatura</b>	Fundamentos de Informática		
<b>Materia</b>	Informática		
<b>Módulo</b>	Materias de FORMACIÓN BÁSICA		
<b>Titulación</b>	Grado en Ingeniería Electrónica Industrial y Automática		
<b>Plan</b>	452	<b>Código</b>	42357
<b>Periodo de impartición</b>	Cuatrimestre 1	<b>Tipo/Carácter</b>	Básica
<b>Nivel/Ciclo</b>	Grado	<b>Curso</b>	1º
<b>Créditos ECTS</b>	6		
<b>Lengua en que se imparte</b>	Castellano		
<b>Profesor/es responsable/s</b>	Félix Miguel Trespaderne Jesús María Zamarreño Cosme		
<b>Datos de contacto (E-mail, teléfono...)</b>	Félix Miguel Trespaderne: <a href="mailto:trespa@eii.uva.es">trespa@eii.uva.es</a> Jesús María Zamarreño Cosme: <a href="mailto:jesusm@autom.uva.es">jesusm@autom.uva.es</a> Tutorías: Consultar la web de la UVa		
<b>Departamento</b>	Ingeniería de Sistemas y Automática		



## 1. Situación / Sentido de la Asignatura

### 1.1 Contextualización

La asignatura Fundamentos de Informática forma parte de las asignaturas del módulo básico (y común) de los Grados en Ingeniería. Esta asignatura es la primera relacionada con la Informática de la titulación.

La procedencia de los estudiantes será fundamentalmente la de alumnos que han cursado el Bachillerato. Dependiendo de las opciones elegidas en los cursos anteriores, habrán tenido diferente grado de contacto con los contenidos desarrollados en esta materia. Así, existirán alumnos que no hayan tenido ninguna formación mientras que otros grupos de alumnos tendrán conocimientos básicos y dispersos según los centros donde hayan cursado asignaturas tales como Tecnologías de la Información y la Comunicación I y II.

Por ello, se pretende ofrecer una formación básica pero sólida, de tal forma que los conocimientos adquiridos sienten las bases para desenvolverse en el resto de materias afines de su formación académica. Se trata de habilitar a los estudiantes para que adquieran las destrezas necesarias para seguir aprendiendo los aspectos relacionados con la Informática, a nivel de manejo de ordenadores y especialmente en el ámbito de la programación.

### 1.2 Relación con otras materias

Conocimientos básicos de Cálculo, Álgebra y Física.

### 1.3 Prerrequisitos

No tiene, pero es recomendable que los estudiantes conozcan los aspectos de lógica impartidos en bachillerato.



## 2. Competencias

### 2.1 Generales

- CG1. Capacidad de análisis y síntesis.
- CG2. Capacidad de organización y planificación del tiempo.
- CG3. Capacidad de expresión oral.
- CG4. Capacidad de expresión escrita.
- CG5. Capacidad para aprender y trabajar de forma autónoma.
- CG6. Capacidad de resolución de problemas.
- CG7. Capacidad de razonamiento crítico/análisis lógico.
- CG8. Capacidad para aplicar los conocimientos a la práctica.
- CG9. Capacidad para trabajar en equipo de forma eficaz.
- CG11. Capacidad para la creatividad y la innovación.

### 2.2 Específicas

- CE3. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.



### 3. Objetivos

- Conocer los fundamentos básicos de la informática y su interrelación con los lenguajes de programación.
- Presentar los conceptos básicos de programación.
- Familiarizarse con las herramientas y procedimientos de una metodología de programación.
- Captar la utilidad de las bases de datos.
- Entender las técnicas de programación imperativa y orientadas a objetos y su utilización en problemas multidisciplinares asociados a la ingeniería.





## 4. Contenidos y/o bloques temáticos

### Bloque 1: Fundamentos de Informática

Carga de trabajo en créditos ECTS:

#### a. Contextualización y justificación

En este bloque se estudia con nivel introductorio aquellos fundamentos de informática complementarios e interrelacionados con los contenidos del Bloque 2 de Fundamentos de Programación. Debe tenerse en cuenta que no existe una asignatura básica que recoja aspectos como la arquitectura de ordenadores, sistemas operativos o bases de datos en la titulación.

#### b. Objetivos de aprendizaje

- Conocer las diversas familias de lenguajes de programación.
- Estudiar las diferencias entre lenguajes compilados y lenguajes interpretados.
- Entender los diferentes sistemas de representación de datos y su interrelación con los tipos usados en los lenguajes de programación.
- Presentar los componentes fundamentales de las arquitecturas Von Neumann y Harvard.
- Comprender el ciclo de instrucción y las interrupciones.
- Estudiar las funciones básicas de un sistema operativo.
- Entender las implicaciones que el uso de las sentencias de un lenguaje de alto nivel tiene sobre los niveles inferiores de la arquitectura de un computador.
- Familiarizarse con los conceptos básicos de las bases de datos.

#### c. Contenidos

- Introducción a la Informática
- Lenguajes de Programación. Paradigmas.
- Sistemas de representación.
- Arquitectura de Computadores.
- Sistemas Operativos.
- Bases de Datos.

#### d. Métodos docentes

Actividades presenciales:

- Lección magistral.
- Resolución de ejercicios

Actividades no presenciales:

- Trabajo individual: Estudio/trabajo personal.

#### e. Plan de trabajo

Para este bloque se estiman 15 horas presenciales distribuidas en 15 horas teóricas en el aula. El tiempo de dedicación no presencial del alumno medio es de unas 22.5 horas.

#### f. Evaluación

La evaluación de este bloque se efectuará en los exámenes oficiales de la asignatura con un peso aproximado del **35%** de la nota final.



## g Material docente

### g.1 Bibliografía básica

- R. E. Bryant, D. R. O'Hallaron. *Computer Systems. A Programmer's Perspective*.  
Pearson 2015. ISBN 978-1292101767
- A. Tanenbaum, T. Austin. *Structured Computer Organization*.  
Pearson 2012. ISBN 978-0273769248
- A. Silberschatz, H. Korth, S. Sudarshan. *Fundamentos de Bases de Datos*.  
McGraw-Hill Education 2014. ISBN 978-8448190330

### g.2 Bibliografía complementaria

- W. Stallings. *Computer Organization and Architecture*.  
Pearson 2015. ISBN 978-0134101613
- J.L. Hennessy, D.A. Patterson. *Computer Architecture: A Quantitative Approach*.  
Morgan Kaufmann 2017. ISBN 978-0128119051

## h. Recursos necesarios

Aula con proyector multimedia y pizarra para sesiones de teoría.  
Plataforma educativa para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.  
Acceso al material bibliográfico recomendado.

## i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Fundamentos de Informática (1.5 ECTS)	Véanse horarios oficiales de la asignatura <a href="https://www.eii.uva.es/">https://www.eii.uva.es/</a>
La temporalización detallada de los contenidos se publicará en el Campus Virtual de la asignatura antes del inicio de la impartición de la asignatura.	



## Bloque 2: Fundamentos de Programación

Carga de trabajo en créditos ECTS: **4,5**

### a. Contextualización y justificación

En este bloque temático se examinará en detalle el paradigma de programación dominante, el imperativo, y se proporcionará una introducción a la programación orientada a objetos.

Para ello, se estudiará el lenguaje de programación multiparadigma C++, centrándose en la parte imperativa del lenguaje.

C++ es un lenguaje más difícil de aprender que otros lenguajes, pero es extremadamente poderoso y ampliamente utilizado en todos los ámbitos de la ingeniería. En cualquier caso, la asignatura se centrará en la parte imperativa, obviando aspectos más avanzados como por ejemplo polimorfismo dinámico. A lo largo del grado o posteriormente en su vida profesional, el alumno o egresado, según sus necesidades, podrá profundizar de forma autodidacta a partir de la base obtenida no sólo en C++, sino en otros lenguajes muy utilizados en ingeniería como Python, MATLAB, etc.

### b. Objetivos de aprendizaje

Entender las técnicas de programación imperativa y orientadas a objetos y su utilización en problemas multidisciplinares asociados a la ingeniería.

Familiarizarse con los entornos de desarrollo modernos de programación para implementar programas sencillos. Presentar los patrones básicos de programación que permiten desarrollar algoritmos sencillos de acuerdo a unas especificaciones dadas.

Conocer los contenedores de datos más habituales y bibliotecas para su uso.

Presentar técnicas para la lectura o el almacenamiento persistente de información.

Entender la importancia de redactar programas siguiendo una regla de estilo y de generar una documentación asociada.

### c. Contenidos

- Variables, operadores, expresiones y sentencias
- Control de flujo
- Funciones
- Estructuras de datos
- Ficheros.

### d. Métodos docentes

Actividades presenciales:

- Clases de teoría-práctica: Lección magistral.
- Prácticas de laboratorio: Estudio de casos.
- Resolución de ejercicios y problemas.
- Aprendizaje basado en problemas y aprendizaje cooperativo.

Actividades no presenciales:

- Trabajo individual: Estudio/trabajo personal.
- Trabajo en grupo: Aprendizaje basado en problemas y aprendizaje cooperativo.

### e. Plan de trabajo

Para este bloque se estiman 45 horas presenciales distribuidas en 15 horas teórico-prácticas y 30 horas de prácticas, todas ellas en laboratorio. El tiempo de dedicación no presencial del alumno medio es de unas 67.5 horas.



## f. Evaluación

La **evaluación teórica** de este bloque se efectuará en los exámenes oficiales de la asignatura con un peso aproximado del **30%** de la nota final.

La **evaluación práctica** de este bloque se efectuará mediante dos procedimientos:

1. Evaluación continua mediante 2 pruebas realizadas en el ordenador en el horario de prácticas de laboratorio. La primera se realizará alrededor de la semana 9 con un peso del **10%** y la última en la última semana lectiva con un peso total del **20%** de la nota final. Las pruebas propondrán resolver problemas de programación de carácter interdisciplinar, típicamente relacionados con algoritmos matemáticos numéricos y problemas de física y química.
2. Evaluación de un trabajo práctico realizado en grupo con un peso del **5%** de la nota final. El trabajo conllevará la consulta de bibliografía y/o manuales de referencia en inglés para su resolución.

La programación temporal se anunciará en el Campus Virtual al inicio del curso.

## g Material docente

### g.1 Bibliografía básica

Fundamentos de Programación en C++. [https://www2.eii.uva.es/fund\\_inf/cpp/](https://www2.eii.uva.es/fund_inf/cpp/)

S. B. Lippman, J. Lajoie, B. E. Moo. *C++ Primer*.  
Addison-Wesley 2012. ISBN-13: 978-0321714114

B. Stroustrup. *Programming: Principles and Practice Using C++*.  
Addison-Wesley Educational Publishers Inc. 2014. ISBN-13: 978-0321992789.

### g.2 Bibliografía complementaria

<http://www.learncpp.com/cpp-tutorial/>

<http://www.cplusplus.com/reference/>

<https://en.cppreference.com/>

## h. Recursos necesarios

Aula con proyector multimedia y pizarra para sesiones de laboratorio y ordenadores para los alumnos.

Plataforma educativa para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.

Acceso al material bibliográfico recomendado.

Compilador de C/C++ (gcc) y entorno de desarrollo Code::Blocks.

## i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 2: Fundamentos de Programación (4.5 ECTS)	Véanse horarios oficiales de la asignatura <a href="https://www.eii.uva.es/">https://www.eii.uva.es/</a>

La temporalización detallada de los contenidos se publicará en el Campus Virtual de la asignatura antes del inicio de la impartición de la asignatura.



**5. Métodos docentes y principios metodológicos**

Actividades presenciales:

- Lección magistral.
- Prácticas de laboratorio: Estudio de casos.
- Resolución de ejercicios y problemas, aprendizaje basado en problemas y aprendizaje cooperativo.

Actividades no presenciales:

- Docencia inversa mediante videos.
- Trabajo individual: Estudio/trabajo personal.
- Trabajo en grupo: Aprendizaje basado en problemas y aprendizaje cooperativo.

**6. Tabla de dedicación del estudiante a la asignatura**

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA <sup>(1)</sup>	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases de teoría	15	Trabajo individual	75
Prácticas de laboratorio	45	Trabajo en grupo	15
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>
			<b>TOTAL</b>
			<b>150</b>

(1) Actividad presencial a distancia es cuando un grupo sigue una videoconferencia de forma síncrona a la clase impartida por el profesor para otro grupo presente en el aula.

**7. Sistema y características de la evaluación**

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Evaluación continua, basada en pruebas parciales	<b>30%</b>	Realizadas en el horario de Laboratorio
Evaluación basada en trabajo en grupo	<b>5%</b>	
Evaluación final	<b>65%</b>	Período de exámenes

**CRITERIOS DE CALIFICACIÓN**

- **Convocatoria ordinaria:**
  - Examen sobre toda la materia, con un peso del **65%**.
  - No existen notas mínimas. La suma de las notas debe igualar o superar el 50%.
  - El **35%** de la nota procede de la evaluación continua y del trabajo en grupo.
- **Convocatoria extraordinaria:**
  - Examen sobre toda la materia, con un peso del **65%**.
  - No existen notas mínimas. La suma de las notas debe igualar o superar el 50%.
  - El **35%** de la nota procede de la evaluación continua y del trabajo en grupo y **no existirán procedimientos de reevaluación** de esta parte.



## 8. Consideraciones finales

La planificación académica podrá sufrir modificaciones de acuerdo con la actualización de las condiciones sanitarias.

En el Campus Virtual de la asignatura se dispondrá de:

- Presentación de la asignatura
- Programa de la asignatura
- Planificación de la asignatura
- Material didáctico
- Enlaces a bibliografía online
- Cronograma de las diferentes actividades de la asignatura.

Con anterioridad a cada práctica de laboratorio, se subirán los guiones de las mismas.