

**Proyecto/Guía docente de la asignatura**

Asignatura	PROGRAMACIÓN		
Materia	INFORMÁTICA		
Módulo	MATERIAS INSTRUMENTALES		
Titulación	GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN		
Plan	460	Código	45004
Periodo de impartición	1 ^{er} . CUATRIMESTRE	Tipo/Carácter	FORMACIÓN BÁSICA
Nivel/Ciclo	GRADO	Curso	1º
Créditos ECTS	6		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	JOSÉ FERNANDO DÍEZ HIGUERA DAVID GONZÁLEZ ORTEGA		
Datos de contacto (E-mail, teléfono...)	DAVID GONZÁLEZ ORTEGA DESPACHO: 2D007 TELÉFONO: 983423000 ext. 5552 E-MAIL: david.gonzalez.ortega@uva.es JOSÉ FERNANDO DÍEZ HIGUERA DESPACHO: 2D079 TELÉFONO: 983 18 55 62 E-MAIL: josdie@tel.uva.es		
Departamento	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		
Fecha de revisión por el Comité de Título	-		



1. Situación / Sentido de la Asignatura

1.1 Contextualización

La asignatura *Programación* pertenece a los planes de estudios de los Grados que se imparten en la Escuela Técnica Superior de Ingenieros de Telecomunicación, siendo la primera asignatura de naturaleza informática a la que se enfrentan los alumnos de dichos Grados.

Esta asignatura es una experiencia de aprendizaje diseñada para enseñar a los alumnos cómo desarrollar programas que puedan resolver problemas, convertir datos, almacenar y recuperar información, ayudar a las personas a comunicarse o hacer cualquier cosa que el programador pueda imaginar. En el fondo, esto se hace cuando el programador escribe las instrucciones para el ordenador en un "lenguaje de programación". En otras palabras, los programadores de ordenadores actúan como traductores entre personas y ordenadores, escribiendo las especificaciones de un programa deseado en un lenguaje que el ordenador pueda entender.

Por último, no hay que perder de vista que, una vez cursada esta asignatura, el alumno debe disponer de una herramienta que tendrá que utilizar en otras asignaturas del Grado y, posiblemente, en su vida profesional.

1.2. Relación con otras materias

Esta asignatura está especialmente relacionada con la asignatura "Fundamentos de Ordenadores y Sistemas Operativos", la cual incluye programación en lenguaje de bajo nivel (ensamblador) y en lenguaje de alto nivel con llamadas al sistema operativo, y con la asignatura "Ingeniería de Sistemas Software", en la que se aplican las técnicas y procedimientos de una metodología de desarrollo software concreta al análisis y diseño de un sistema software en el ámbito de los servicios de telecomunicación.

1.3. Prerrequisitos

El alumno que curse esta asignatura ha de poseer unos conocimientos básicos de informática a nivel usuario. En lo referente a programación, y siendo la primera asignatura de los Grados que aborda dicha materia, se parte de la base de que el alumno no tiene conocimientos previos de la misma.

2. Competencias

2.1 Generales

- GB1. Capacidad de razonamiento, análisis y síntesis.
- GB2. Capacidad para relacionar conceptos y adquirir una visión integrada, evitando enfoques fragmentarios.
- GB3. Capacidad de toma de decisiones en la resolución de problemas básicos de ingeniería de telecomunicación, así como identificación y formulación de los mismos.
- GB4. Capacidad para trabajar en grupo, participando de forma activa, colaborando con sus compañeros y trabajando de forma orientada al resultado conjunto, y en un entorno multilingüe.
- GB5. Conocimiento de materias básicas, científicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías.
- GBE3. Capacidad para resolver problemas con iniciativa, creatividad y razonamiento crítico.
- GC1. Capacidad de organización, planificación y gestión del tiempo.
- GC2. Capacidad para comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
- GC3. Trabajar en cualquier contexto, individual o en grupo, de aprendizaje o profesional, local o internacional, desde el respeto a los derechos fundamentales, de igualdad de sexo, raza o religión y los principios de accesibilidad universal, así como la cultura de paz.

2.2. Específicas

- B2. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

3. Objetivos

Al finalizar la asignatura el alumno deberá ser capaz de:

- Conocer los conceptos relacionados con la programación.
- Aplicar las técnicas y procedimientos de una metodología de programación de un sistema software.
- Codificar y probar dicho sistema, aplicando técnicas de programación orientada a procesos y a datos.
- Codificar, poner a punto y ejecutar programas sencillos en lenguaje C.
- Diseñar algoritmos sencillos basados en los esquemas de recorrido y búsqueda.
- Autoevaluar el trabajo realizado e identificar los propios errores y aspectos a mejorar.
- Aprender de forma autónoma:
 - Localizar y asimilar una determinada información a partir de su referencia.
 - Autoevaluarse o evaluar a otros a partir de unos criterios dados (Evaluación).
 - Identificar los propios errores.
 - Buscar información relevante para una tarea especificada.

4. Niveles de Aprendizaje

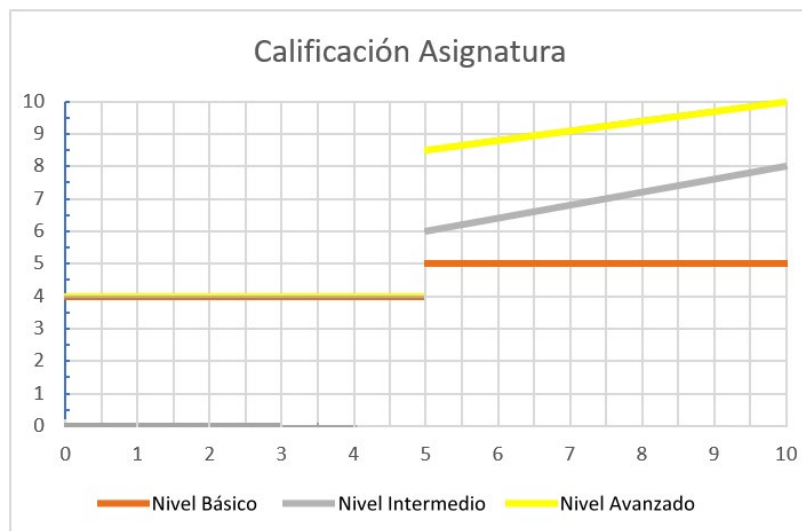
En el presente curso, la asignatura se articula en tres niveles de aprendizaje, a semejanza de los cursos de idiomas, ya que el objetivo principal es el aprendizaje de un lenguaje (de programación): *Nivel Básico*, *Nivel Intermedio* y *Nivel Avanzado*. Con este esquema se pretende que sea el alumno el que decida los conocimientos y habilidades que quiere aprender sobre programación de ordenadores y/o la calificación que desea conseguir para su expediente. En este sentido, las calificaciones que se pueden obtener superando las pruebas de cada uno de los niveles, y la duración de cada nivel, se muestran en la siguiente tabla:

Nivel	Duración	Calificación
Básico	8 semanas	5.0
Intermedio	4 semanas	[6.0 - 8.0]
Avanzado	3 semanas	[8.5 - 10.0]

- Al finalizar las semanas correspondientes al nivel básico, se publicarán las calificaciones obtenidas en el nivel básico. Aquellos alumnos que superen este nivel con una nota igual o superior a 5.0 **ya obtienen una calificación de 5.0 como nota de la asignatura**. A partir de ese momento, será potestad de cada alumno decidir si quiere continuar con el nivel intermedio. En caso contrario, la nota definitiva será de 5.0 puntos (aprobado).
- Al finalizar las semanas correspondientes al nivel intermedio, se publicarán las calificaciones de aquellos alumnos que hayan decidido cursar el nivel intermedio. Aquellos alumnos que superen este nivel con una nota igual o superior a 5.0 **podrán optar a una calificación entre 6.0 y 8.0**, en función de los resultados de la evaluación de este nivel. A partir de ese momento, será potestad de cada alumno decidir si quiere continuar con el nivel avanzado. En caso contrario, la nota definitiva será la obtenida en las pruebas de este nivel, y siempre en el intervalo [6.0, 8.0].
- Por último, al finalizar las semanas correspondientes al nivel avanzado, se publicarán las calificaciones de aquellos alumnos que hayan decidido cursar el nivel avanzado. Aquellos alumnos que superen este nivel con una nota igual o superior a 5.0 **podrán optar a una calificación entre 8.5 y 10.0**, en función de los resultados de la evaluación de este nivel.

En la siguiente gráfica se pueden observar las escalas de calificaciones correspondientes a cada uno de los niveles de aprendizaje.

NOTA: Como se indica más adelante, en la sección Criterios de Evaluación, *si un alumno no alcanza los requisitos mínimos de cada elemento de evaluación, su calificación final en la asignatura será el mínimo entre el valor calculado según la ponderación descrita en la tabla y 4.0*. En la gráfica aparece la calificación de 4 puntos, aunque puede ser inferior.



5. Métodos docentes y principios metodológicos

Siguiendo una de las recomendaciones de la Dirección de la ETSIT en cursos anteriores, el flujo de trabajo semanal de esta asignatura está edificado sobre los cimientos del modelo pedagógico denominado *Flipped Classroom* (FC), complementado con técnicas de *Just-in-time Teaching* y *ConcepTest*.

- *Flipped Classroom* (o *clase inversa*) es “un modelo pedagógico que transfiere el trabajo de determinados procesos de aprendizaje fuera del aula y utiliza el tiempo de clase, junto con la experiencia del docente, para facilitar y potenciar otros procesos de adquisición y práctica de conocimientos dentro del aula”¹.
- *Just-in-time (JIT) Teaching* (o *enseñanza justo a tiempo*) es una estrategia pedagógica que utiliza la retroalimentación entre las actividades de aula y el trabajo que el alumnado hace en casa para preparar las sesiones que se dan en el aula. Los objetivos son: aumentar el aprendizaje durante el tiempo que se está en el aula, fomentar la motivación de los alumnos, animar a los estudiantes a que se preparen las clases y permitir al profesor encontrar las actividades más adecuadas para las necesidades de sus estudiantes.
- *ConcepTests* (o *pruebas de concepto*) son pruebas cortas, informales y dirigidas que se administran durante la clase para ayudar a los instructores a evaluar si los estudiantes comprenden los conceptos clave. Se pueden usar tanto para evaluar los conocimientos previos de los estudiantes (entrar en un curso o unidad) como la comprensión del contenido del curso o unidad actual. Por lo general, estas pruebas consisten en una a cinco preguntas de opción múltiple. Se les pide a los estudiantes que seleccionen la mejor respuesta y la presenten levantando sus manos, sosteniendo una tarjeta de color asociada con una opción de respuesta, o usando un dispositivo de control remoto para ingresar su respuesta.

Como ya se ha comentado en la sección 4, la asignatura se estructura en niveles de aprendizaje. Cada nivel se compone de varias etapas (por lo general, de una semana de duración), que a su vez se articulan en una serie de tramos de diferentes actividades. A continuación, se describe la hoja de ruta de una etapa estándar (sin jornadas no lectivas), con la enumeración de los tramos a cubrir.

I. Briefing

Tipo de actividad: no presencial

Al comienzo de cada tramo, y a través del Campus Virtual de la asignatura, el profesor proporciona a los alumnos la hoja de ruta correspondiente a dicho tramo, con la siguiente información:

- Fecha de inicio y final de tramo.
- Objetivos de aprendizaje.
- Enlaces a los recursos necesarios para conseguir los objetivos de aprendizaje previstos: textos.
- Actividades presenciales y de trabajo individual no presencial a realizar durante el tramo, indicando cuáles de ellas son formativas, y cuáles tendrán una calificación sumativa.
- Criterios específicos de calificación de las actividades evaluables.

II. Periplo a través de los recursos

Tipo de actividad: no presencial	No evaluable
----------------------------------	--------------

En este tramo el alumno debe utilizar los recursos proporcionados por el profesor, y/o leyendo PDFs interactivos, tutoriales y otros tipos de documentación escrita, relacionados con el material que se abordará durante la próxima sesión de clase.

III. Aplicación de la enseñanza Just-in-time

Tipo de actividad: no presencial	Tipo de evaluación: diagnóstica
----------------------------------	---------------------------------

Para esta actividad el profesor habrá desarrollado previamente un conjunto de preguntas efectivas que se publicarán en el Campus Virtual para que los estudiantes respondan antes de la próxima clase. Estas preguntas son generalmente de opción múltiple, y requieren que los estudiantes hayan completado el tramo anterior. Después de la fecha límite de publicación, pero siempre antes de que comience la clase, el profesor examina las respuestas de los estudiantes. En la mayoría de los casos, el profesor puede utilizar esta revisión para hacer ajustes en las actividades de aula previstas. Si el profesor cree que el alumnado ha dominado un tema, puede reducir o eliminar la discusión de dicho tema durante la clase. De igual manera, si las respuestas muestran que los alumnos tienen dificultades concretas, estas serán vistas durante la clase con más detenimiento.

¹ ¿Qué es Flipped Classroom? [https://www.theflippedclassroom.es/what-is-innovacion-educativa/#:~:text=El%20Flipped%20Classroom%20\(FC\)%20es,de%20conocimientos%20dentro%20del%20aula.](https://www.theflippedclassroom.es/what-is-innovacion-educativa/#:~:text=El%20Flipped%20Classroom%20(FC)%20es,de%20conocimientos%20dentro%20del%20aula.)

IV. Sesión de clase

A. Procedimiento de desarrollo de la sesión

En circunstancias normales, y atendiendo a las características específicas de esta asignatura, la sesión de clase en aula se desarrolla mediante el siguiente procedimiento: el profesor conecta su portátil al proyector del aula, con el fin de que todos los alumnos puedan visualizar en la pantalla del aula una presentación (*PowerPoint*, *Google Colab*, *H5P*, etc.), el desarrollo de un ejercicio en el portátil del profesor utilizando el entorno de desarrollo del lenguaje de programación utilizado, y cualquier otro documento o aplicación que el profesor desee mostrar. En otras palabras, el profesor **comparte** la pantalla de su portátil con los alumnos a través del proyector y la pantalla de proyección instalados en el aula. Con cierta frecuencia, el profesor puede 'invitar' a algunos alumnos a salir a la pizarra para realizar un ejercicio, lo que en Programación se traduce en utilizar el ordenador del profesor para editar el código, de forma que el resto de la clase pueda seguir el desarrollo, sugiriendo soluciones, detectando errores en el código, etc., en lo que se podría calificar como una actividad de *aprendizaje cooperativo*, ya que todos los alumnos cooperan en la resolución del problema.

Por otro lado, y en el mejor de los escenarios en el que todos los alumnos puedan encontrar acomodo en el aula, se plantea el problema de que los alumnos más alejados de la pantalla de proyección pueden tener serias dificultades para visualizar correctamente los contenidos allí proyectados. Como solución a este inconveniente, durante las sesiones de aula se utilizará la aplicación de videoconferencia *Blackboard Collaborate* para que los alumnos que tengan dificultades audiovisuales para seguir la clase lo puedan hacer a través de un portátil que necesariamente tendrán que llevar a clase. *[Blackboard Collaborate es una plataforma que permite generar espacios de comunicación en forma de salas virtuales, donde varios participantes se reúnen y pueden compartir diferentes aplicaciones: presentaciones, escritorio, pizarra virtual, etc.]*. De esta forma la pantalla del profesor se compartirá simultáneamente a través del proyector y de *Blackboard Collaborate*. Este procedimiento permite también mantener el aprendizaje cooperativo en el desarrollo de ejercicios en clase, ya que el salir a la pizarra, en este contexto, se traduce en que un alumno comparta la ventana del entorno de desarrollo con el resto de la clase a través de *Blackboard Collaborate*. Así mismo, este procedimiento permite que alumnos con riesgo, o alumnos que no puedan acceder al aula por exceso de aforo, puedan seguir la clase de forma telepresencial, e interactuar con sus compañeros y con el profesor en las mismas condiciones que los alumnos presentes en el aula. Huelga decir que en ningún caso se plantea la utilización de dispositivos de captación de imagen (ya que lo que interesa al alumno es el contenido y la explicación del profesor y la interacción con sus compañeros), y en ningún caso se va a grabar la sesión de clase compartida a través de la plataforma *Blackboard Collaborate*.

Y lo que es más importante, este procedimiento es igualmente válido en el caso de que las circunstancias aconsejen u obliguen a conmutar toda la docencia a modo telepresencial, sin tener que efectuar cambio alguno, salvo la ubicación física de los alumnos.

B. Desarrollo de la sesión

Como se ha comentado en la sección 3, el profesor, después de analizar los resultados de la enseñanza *Just-in-time*, habrá planificado la sesión con recursos y actividades que profundicen en conceptos deficientemente asimilados por los alumnos. Una vez desplegados dichos recursos, el profesor utilizará la técnica de *ConcepTest* para recabar nueva información sobre la asimilación grupal de conceptos, al menos una vez durante la sesión (evaluación diagnóstica), con aplicaciones del tipo *Kahoot*, *Socrative*, etc., y al final de cada sesión (evaluación sumativa), mediante un cuestionario de Moodle realizado con la app para móviles de dicha plataforma (se plantea esta alternativa ya que es posible que algún alumno no disponga de portátil, pero es casi seguro que todos tienen un teléfono móvil).

V. Sesión de laboratorio

A. Procedimiento de desarrollo de la sesión

En circunstancias normales, y atendiendo a las características específicas de esta asignatura, la sesión de laboratorio se desarrolla mediante el siguiente procedimiento: cada alumno ocupa un puesto de trabajo, y el profesor monitoriza el trabajo de los alumnos y atiende puntual y localmente (sobre el ordenador del alumno) las dudas o problemas que les puedan surgir en el desarrollo de las actividades planificadas para la sesión.

En circunstancias excepcionales, como las vividas recientemente durante la pandemia, las medidas de distanciamiento interpersonal impiden tanto la movilidad de los alumnos en el laboratorio (deben permanecer durante toda la sesión en su puesto de trabajo), como la atención personalizada del profesor. En este escenario, la utilización de la plataforma *Blackboard Collaborate* durante toda la sesión de laboratorio permitirá superar estos obstáculos en la comunicación alumno-alumno y alumno-profesor. En el caso de que un alumno solicite ayuda, el profesor habilitará una sala virtual para atender a dicho alumno, que en la sala podrá compartir pantalla o ventana de aplicación y exponer su problema al profesor). Por otro lado, y en determinadas prácticas, el profesor habilitará salas virtuales para que los alumnos puedan cooperar virtualmente, sin tener que abandonar su puesto de trabajo. Además, en el caso de que el profesor necesite explicar algún concepto relacionado con la práctica, puede utilizar la plataforma para compartir su pantalla con los alumnos, no siendo necesaria la utilización de proyector alguno, que en la actualidad no está disponible en el laboratorio.

Y lo que es más importante, y al igual que en el caso de las sesiones en aula, este procedimiento es igualmente válido en el caso de que las circunstancias aconsejen u obliguen a conmutar toda la docencia a modo telepresencial, sin tener que efectuar cambio alguno, salvo la ubicación física de los alumnos.

B. Desarrollo de la sesión de laboratorio

El objetivo global de las prácticas de laboratorio es la elaboración de un proyecto integrador que implante la mayor parte de los conocimientos y habilidades de programación que el alumno debe adquirir en los diferentes niveles de aprendizaje. En este sentido, en cada sesión de trabajo en el laboratorio se proporcionarán al alumno las especificaciones de una práctica guiada, en la que tendrá que ir integrando en el proyecto nuevas funcionalidades basadas en los conocimientos y habilidades de la etapa correspondiente.

Como soporte de las prácticas se utiliza el servidor de control de versiones *GitLab* de la E.T.S. Ingenieros de Telecomunicación (<https://github.tel.uva.es/>). Este soporte facilita diferentes funciones:

- Control de versiones: los alumnos pueden registrar las sucesivas versiones de su proyecto, con la posibilidad de recuperar versiones anteriores, y la potencialidad de disponer de un repositorio remoto al que acceder desde diferentes máquinas (ordenador personal, ordenadores del laboratorio, etc.)
- Seguimiento del proyecto: este mecanismo permite al profesor monitorizar la evolución del código desarrollado por cada alumno.
- Control *anti-cyranos*: al comienzo de cada sesión de laboratorio, ya sea presencial o telepresencial, el alumno debe lanzar un *demonio* o servicio que periódicamente realiza *commits* (actualizaciones de versión) en segundo plano, lo que facilita el control de versiones, ya que el alumno no se tiene que preocupar de preparar y ejecutar la actualización del código de su proyecto en GitLab. Además de las actualizaciones de código, el demonio también envía información sobre la máquina en la que se está realizando la práctica, la IP utilizada, y otros datos que permitan detectar la posible (e indeseable) intervención de "cyranos", esto es, personas que dictan las soluciones al alumno o incluso llegan a suplantarlos en actividades telepresenciales.

Una vez finalizada la sesión de prácticas, el alumno debe entregar la última versión estable en el correspondiente buzón/taller del Campus Virtual.

Nota: las prácticas guiadas están diseñadas de forma que puedan ser realizadas durante la sesión de laboratorio. El código susceptible de coevaluación es el que se entrega al final de la sesión; no obstante, el alumno que no haya completado la práctica al final de dicha sesión, deberá completarla como actividad no presencial, ya que en la siguiente sesión la práctica guiada tomará como punto de partida la versión completa de la etapa anterior.

VI. Análisis de la etapa

Tipo de actividad: no presencial	Tipo de evaluación: formativa
----------------------------------	-------------------------------

En este último tramo se analiza y evalúa el resultado del aprendizaje realizado durante la etapa. Para ello, la entrega de cada alumno es coevaluada por al menos tres alumnos seleccionados de forma aleatoria, utilizando la herramienta "Taller" de Moodle. Los beneficios de esta tarea son:

- El alumno recibe una valiosa realimentación de sus compañeros, ya que la calificación no se limita a una simple nota numérica.
- El alumno puede contrastar su solución con la de los compañeros a los que evalúa, e incluso comprobar cómo otro compañero ha resuelto alguna funcionalidad que él no ha sido capaz de resolver.

6. Contenidos y/o bloques temáticos

En consonancia con la sección Niveles de Aprendizaje, los conocimientos y habilidades que el alumno debe adquirir se articulan en tres bloques temáticos, correspondientes a los tres niveles de aprendizaje.

Los objetivos de aprendizaje describen en detalle todo lo que el alumno va a aprender durante este curso. Es importante que el alumno los tenga presentes desde el primer momento, aunque el profesor le irá recordando los objetivos que están implicados en las diferentes actividades del curso.

6.1. Características comunes a todos los bloques

a. Métodos docentes

- Clase expositiva participativa.
- Resolución de problemas.
- Desarrollo del Proyecto en el laboratorio.
- Aprendizaje basado en proyecto.

Para más información, consultar la sección *Métodos docentes y principios metodológicos*.

b. Plan de trabajo

El Plan de Trabajo de la asignatura, donde se describe la planificación detallada, incluyendo fechas de las prácticas por grupo, se entregará al comienzo de la asignatura.

c. Evaluación

- Pruebas de conocimientos básicos.
- Evaluación de la versión final del Proyecto correspondiente a cada nivel.
- Coevaluación de la versión del Proyecto correspondiente a cada nivel.

Para más información, consultar la sección **Sistema y características de la evaluación**.

La normativa y criterios de evaluación específicos se entregarán al comienzo de la asignatura.

d. Material docente

La bibliografía propuesta para esta asignatura tiene como objeto servir de soporte al aprendizaje de los conceptos y del lenguaje de programación de la asignatura, por lo que estudiarse alguno (¡o todos!) los libros, no tiene sentido por sí mismo.

Sin embargo, recomendamos el uso de alguno de los libros mencionados en el enlace a la [Plataforma Leganto](#) de la Biblioteca de la UVA con la bibliografía recomendada. (en el orden en el que aparecen). Existen ejemplares en la biblioteca y todos ellos tratan el mismo tema, con diferentes enfoques, ejemplos y orden de exposición.

§ Otros recursos telemáticos

Los recursos documentales, audiovisuales y de hipermedia elaborados para la implantación de la clase inversa (*flipped classroom*), se enlazarán directamente en las secciones correspondientes del Campus Virtual.

e. Recursos necesarios

- Aula con proyector multimedia y pizarra para sesiones de discusión.
- Laboratorio de prácticas, con un ordenador por alumno, para las sesiones de laboratorio. Cada ordenador debe contar con el entorno de desarrollo para el lenguaje C que se va a utilizar.
- Plataforma educativa para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.
- Plataforma de videoconferencia para compartir pantalla y/o aplicaciones en sesiones telepresenciales.

En esta asignatura se utilizan herramientas docentes online para la docencia y la evaluación. En caso de un transcurso normal de la docencia estarán disponibles las aulas informáticas del centro. En caso de una afección por medidas sanitarias especiales, el alumno debe contar con medios informáticos y telemáticos suficientes para interactuar con el Campus Virtual y con los sistemas de videoconferencia.

6.2. Desglose por bloques

Bloque 1: Nivel Básico

Carga de trabajo en créditos ECTS: 3.2

a. Contextualización y justificación

En este primer bloque se introducen los conceptos básicos sobre los diferentes métodos para el desarrollo de aplicaciones informáticas, se presenta C, el lenguaje de programación que se utilizará para la implantación de los programas desarrollados, y se proporcionan los elementos básicos de la programación estructurada. El bloque se completa con casos de estudio y con la aplicación al proyecto integrador de los métodos aprendidos.

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con los **tipos de datos** elementales y sus **operaciones** (enteros, caracteres y reales):
 - Escribir los tipos de datos elementales y las operaciones que actúan sobre ellos.
 - Escribir la declaración de datos de cualquiera de los tipos elementales.
 - Indicar el código ASCII de cualquier carácter, con la ayuda de la tabla correspondiente.
- En relación con los **punteros**:
 - Comprender el concepto de puntero y sus diferencias con una variable simple.
 - Escribir el código necesario para acceder al dato apuntado por un puntero.
- En relación con las **sentencias básicas de asignación y de entrada/salida**:
 - Describir el funcionamiento de las sentencias básicas.
 - Predecir el resultado de una secuencia de sentencias básicas.
 - Codificar una tarea convenientemente especificada, utilizando la secuencia de sentencias básicas adecuada.
- En relación con los **archivos**:
 - Explicar el concepto de archivo, para qué sirve, y cuáles son las operaciones típicas sobre archivos de texto (crear, abrir, leer, escribir, preguntar por fin de archivo y cerrar).
 - Escribir las sentencias necesarias para realizar las operaciones básicas con un archivo de texto.
 - Escribir las sentencias necesarias para determinar el tipo de error que se ha producido al realizar una operación con un archivo de texto.
- En relación con las **sentencias de control**:
 - Conocer y usar los operadores que se utilizan para construir expresiones lógicas.
 - Usar las sentencias de selección `if`, `if-else` para elegir entre varias acciones alternativas.
 - Utilizar la sentencia de selección múltiple `switch` para escoger entre muchas alternativas de acción.
 - Utilizar las etiquetas `case` para identificar las acciones alternativas en las sentencias `switch`.
 - Ejecutar sentencias repetidamente con la sentencia repetitiva `for`.
 - Usar la sentencia de repetición `while` para ejecutar sentencias repetidamente.
 - Usar la sentencia de repetición `do while` para ejecutar sentencias repetidamente.
- En relación con las **funciones**:
 - Definir el concepto de función y su utilidad.
 - Definir los conceptos: cabecera de función, parámetros formales, variables locales, resultado de la función, llamada a función, parámetros reales, paso de parámetros.
 - Describir la diferencia entre paso de parámetros por valor o por referencia.
 - Codificar convenientemente una llamada a función, pasando correctamente los parámetros.
 - Codificar en forma de función una tarea convenientemente especificada, estableciendo adecuadamente los parámetros necesarios.

- Proponer una organización en bloques (funciones) de una aplicación determinada.
- En relación con el **entorno de desarrollo**:
 - Definir los conceptos de compilación, construcción y ejecución.
 - Realizar las operaciones necesarias para crear/abrir un proyecto y añadir y eliminar elementos a un proyecto.
 - Realizar las operaciones necesarias para editar, compilar, montar y ejecutar un programa, y localizar las carpetas donde están los archivos generados en cada uno de los pasos.
 - Interpretar adecuadamente los mensajes de error de compilación y corregir el error de compilación correspondiente.
 - Describir las funcionalidades básicas del depurador.
 - Realizar correctamente las operaciones básicas del depurador (insertar un punto de parada, ejecutar paso a paso y visualizar valores de variables).
 - Identificar y subsanar los errores de ejecución de un programa, utilizando adecuadamente el depurador.

c. Contenidos

- Introducción a la Programación.
- Introducción al Lenguaje C
- Funciones.
- Sentencias de Control.
- Vectores numéricos y cadenas de caracteres.

Bloque 2: Nivel Intermedio

Carga de trabajo en créditos ECTS:

1.6

a. Contextualización y justificación

El bloque 2 se centra en la introducción de los conceptos de puntero, paso de parámetros por dirección a una función, tipos de datos estructurados, registros, archivos binarios, vectores multidimensionales y vectores de registros, estructuras de datos dinámicas, y asignación dinámica de memoria. El bloque se completa con casos de estudio y con la aplicación al proyecto integrador de los métodos aprendidos.

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con los **punteros**:
 - Escribir el código necesario para recorrer un vector o cadena de caracteres utilizando un puntero.
 - Implantar el paso de parámetros por dirección a una función utilizando punteros.
- En relación con los **tipos de datos estructurados**:
 - Describir las estructuras de datos fundamentales, y las operaciones típicas sobre ellas.
 - Escribir la declaración de una estructura de datos convenientemente especificada.
 - Escribir el código necesario para acceder a un elemento o conjunto de elementos de una estructura de datos.
 - Elegir la estructura de datos más adecuada para una aplicación determinada.
- En relación con los **archivos binarios**:
 - Recordar el concepto de archivo, para qué sirve, y cuáles son las operaciones típicas sobre archivos (crear, abrir, leer, escribir, preguntar por fin de archivo y cerrar).
 - Escribir las sentencias necesarias para realizar las operaciones básicas con archivos binarios.
 - Escribir las sentencias necesarias para determinar el tipo de error que se ha producido al realizar una operación con un archivo binario.
- En relación con la **asignación dinámica de memoria**:
 - Asignar y liberar memoria dinámicamente para diferentes tipos de datos.
 - Crear y redimensionar vectores dinámicos.
- En relación con los **vectores dinámicos**:

- Escribir la declaración de un vector dinámico.
- Escribir el código necesario para asignar memoria a un vector dinámico.
- Escribir el código necesario para reasignar memoria a un vector dinámico.
- Escribir el código necesario para liberar la memoria asignada un vector dinámico.

c. Contenidos

- Registros
- Archivos Binarios.
- Estructuras Dinámicas de Datos.

Bloque 3: Nivel Avanzado

Carga de trabajo en créditos ECTS: 1.2

a. Contextualización y justificación

El bloque 3 se centra en la introducción a la gestión de bases de datos desde aplicaciones C. El bloque se completa con casos de estudio y con la aplicación al proyecto integrador de los nuevos conceptos aprendidos.

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con las **bases de datos**:
 - Entender el concepto de base de datos, y cómo se estructura.
 - Escribir las sentencias necesarias para crear, abrir y cerrar una base de datos sencilla (*SQLite*).
 - Escribir las sentencias necesarias para realizar las operaciones básicas sobre una base de datos (crear tablas, insertar datos, seleccionar datos).

c. Contenidos

- Introducción a Bases de datos.



7. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Nivel Básico	3.2	Semanas 1 a 8
Bloque 2: Nivel Intermedio	1.6	Semanas 9 a 12
Bloque 3: Nivel Avanzado	1.2	Semanas 13 a 15



8. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA ⁽¹⁾	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Sesiones de aula	20	Visionado y/o lectura de recursos didácticos	35
Prácticas guiadas en laboratorio	40	Cuestionarios de enseñanza JIT	10
		Coevaluación de versiones del proyecto	15
		Horas-extra para el desarrollo del proyecto ⁽²⁾	30
Total presencial	60	Total no presencial	90
TOTAL presencial + no presencial			150

- (1) Actividad presencial a distancia es aquella en la que un grupo de alumnos sigue una videoconferencia de forma síncrona a la clase impartida por el profesor.
- (2) Estas horas de trabajo no presencial sólo se utilizarán para completar las prácticas guiadas semanales, en el caso de que el alumno no haya conseguido finalizarlas durante la sesión de laboratorio. De esta forma se mantiene el compromiso adquirido con la CGCC de aligerar en lo posible la carga no presencial de los alumnos.

9. Sistema y características de la evaluación

La evaluación de la asignatura se articula en tres elementos: cuestionarios de conocimientos básicos, proyecto integrador y coevaluación de proyectos.

9.1. Cuestionarios de Conocimientos Básicos

Esta asignatura dispone de un banco de preguntas residente en el Campus Virtual. A partir de este banco de preguntas se elaboran una serie de 11 cuestionarios de Conocimientos Básicos:

- SIETE** para el Nivel Básico:
 - CB01: Introducción a la Programación
 - CB02: Introducción al Lenguaje C
 - CB03: Funciones
 - CB04: Cadenas de Caracteres
 - CB05: Sentencias Selectivas
 - CB06: Sentencias Iterativas
 - CB07: Vectores
- TRES** para el Nivel Intermedio:
 - CB08: Registros
 - CB09: Archivos Binarios
 - CB10: Estructuras Dinámicas de Datos
- UNO** para el Nivel Avanzado
 - CB11: Base de Datos SQLite

Los alumnos pueden acceder y practicar con las preguntas de este banco de preguntas a través de los **cuestionarios de aprendizaje** disponibles durante todo el curso.

Cada cuestionario de Conocimientos Básicos consta de 10 preguntas, seleccionadas aleatoriamente del banco de preguntas. Para superar un cuestionario de conocimientos básicos, el alumno debe responder correctamente al menos **a 8 de las 10 preguntas**.

Para superar cada cuestionario, el alumno dispone de cuatro intentos durante el curso, más dos intentos adicionales en la convocatoria ordinaria y otros dos en la convocatoria extraordinaria. **La calificación de cada cuestionario será el resultado de promediar las calificaciones de todos los intentos realizados.**

- **Nivel Básico:** para promediar con el resto de los elementos de evaluación, el alumno debe responder correctamente al menos a 8 de las 10 preguntas de los 7 primeros cuestionarios.
- **Nivel Intermedio:** solamente incluye tres cuestionarios de Conocimientos Básicos. Para promediar con el resto de los elementos de evaluación, el alumno debe responder correctamente al menos a 8 de las 10 preguntas de los 10 primeros cuestionarios: los 7 del nivel básico, más los 3 correspondientes a este nivel.
- **Nivel Avanzado:** solamente incluye un cuestionario de Conocimientos Básicos. Para promediar con el resto de los elementos de evaluación, el alumno debe responder correctamente al menos a 8 de las 10 preguntas de los 11 cuestionarios: los 7 del nivel básico, más los 3 del nivel intermedio y el correspondiente a este nivel.

La calificación de conocimientos básicos correspondiente a cada nivel de aprendizaje se calcula como promedio de las calificaciones obtenidas en los conocimientos básicos correspondientes a dicho nivel.

9.2. Proyecto Integrador

A lo largo del curso, y durante las horas presenciales en laboratorio, el alumno desarrollará un proyecto informático que se irá actualizando con la incorporación de los conceptos aprendidos en cada semana (ver sección **Desarrollo de la sesión de laboratorio**). En la versión de cada nivel se incorporarán los nuevos elementos de programación vistos en las semanas correspondientes. En cada sesión de laboratorio, el alumno realizará la práctica guiada con la tutorización del profesor, y entregará la versión desarrollada en el buzón correspondiente del Campus Virtual, para que dicho envío sea coevaluado por al menos tres alumnos de forma anónima. Esta evaluación será revisada por el profesor en los siguientes casos:

- a) Previa petición de los alumnos que consideren, razonadamente, que su evaluación no es correcta, o
- b) En el caso de que se detecten serias discrepancias en las calificaciones de los evaluadores.

§ Ampliación individual

Con el objeto de validar el proyecto elaborado y entregado por cada alumno, en la última semana de cada nivel (semanas 8, 12 y 15, respectivamente), el alumno desarrollará una nueva versión del proyecto. En esta nueva versión no se incorporarán nuevos conceptos, sino que se realizarán tareas similares a las realizadas en versiones anteriores. Cada tarea a desarrollar en esta versión ampliada tendrá asignado un cierto número de puntos, y para demostrar que el alumno conoce la estructura y funciones de la última versión entregada, y que, por tanto, es autor de la misma (ya que tiene que realizar tareas similares), para superar esta prueba tendrá que conseguir al menos el 60% de los puntos totales. El baremo será:

Porcentaje de puntos obtenidos	Calificación
Menos del 60%	0.0
60%	5.0
Entre 60% y 100%	0.125p - 2.5
100%	10.0

La nota final del proyecto será, en cada nivel:

Elemento de evaluación	Porcentaje	Nota mínima
Coevaluación de las diferentes versiones	35%	5/10
Ampliación individual	65%	5/10

Convocatorias para cada nivel:

- El alumno que opte por el **nivel básico** y no apruebe el proyecto en la entrega de la semana 8, podrá completarlo y/o modificarlo en las siguientes sesiones de laboratorio, y volver a intentarlo en las entregas de las semanas 12 y 15, en la de la convocatoria ordinaria, y en la de la convocatoria extraordinaria. En total, 5 intentos.
- El alumno que opte por el **nivel intermedio** y no apruebe el proyecto en la entrega de la semana 12, podrá completarlo y/o modificarlo en las siguientes sesiones de laboratorio, y volver a intentarlo en las entregas de la semana 15, en la de la convocatoria ordinaria, y en la de la convocatoria extraordinaria. En total, 4 intentos.
- El alumno que opte por el **nivel avanzado** y no apruebe el proyecto en la entrega de la semana 15, podrá completarlo y/o modificarlo, y volver a intentarlo en las entregas de la convocatoria ordinaria, y en la de la convocatoria extraordinaria. En total, 3 intentos.

Para promediar con el resto de los elementos de evaluación, el proyecto debe obtener una calificación mínima de 5 sobre 10.

9.3. Coevaluación

La calificación de este elemento vendrá dada por las notas obtenidas por el alumno en las actividades de coevaluación y autoevaluación de las sucesivas versiones del proyecto. La calificación correspondiente a cada nivel de aprendizaje se calcula como **promedio de las calificaciones obtenidas en las coevaluaciones correspondientes a dicho nivel**.

10. Criterios de Evaluación

10.1. Convocatoria

Siguiendo las recomendaciones de la Dirección del Centro, al comienzo del curso se ofrecerá a los alumnos veteranos la posibilidad de convalidar los elementos de evaluación que hayan superado en el curso anterior, *prorrogando a este curso los mismos criterios utilizados para las convocatorias ordinaria y extraordinaria*.

10.2. Evaluación Continua

Debido a la estructuración de la asignatura en tres niveles de aprendizaje, el alumno tiene la posibilidad de superar la asignatura sin necesidad de acudir a la convocatoria ordinaria, obteniendo las puntuaciones requeridas en la evaluación continua.

ELEMENTO DE EVALUACIÓN	PESO EN LA NOTA FINAL	OBSERVACIONES
Cuestionarios de Conocimientos básicos	30%	Para superar cada cuestionario de conocimientos básicos el alumno dispone de cuatro intentos durante el curso, más dos intentos adicionales en las convocatorias ordinaria y extraordinaria.
Proyecto integrador	60%	Para promediar con el resto de los elementos de evaluación, el proyecto debe obtener una calificación mínima de 5 sobre 10.
Coevaluación	10%	La calificación de este elemento vendrá dada por las notas obtenidas por el alumno en las actividades de coevaluación y autoevaluación de las versiones del proyecto. Nota mínima: 5 sobre 10.

Si un alumno no alcanza los requisitos mínimos descritos en la tabla anterior, su calificación final en la asignatura será el mínimo entre el valor calculado según la ponderación descrita en la tabla y 4.0.

10.3. Evaluación en la Convocatoria Ordinaria

La Convocatoria Ordinaria se realiza en la fecha prevista en el calendario de exámenes. Esta convocatoria está indicada para:

1. Alumnos que no han aprobado alguno de los niveles en la Evaluación Continua.
2. Alumnos que, habiendo aprobado alguno de los niveles en la evaluación continua, deseen mejorar su calificación presentándose a un nivel superior al aprobado. En este caso, si un alumno no supera el nivel al que se presenta en esta convocatoria, mantiene la calificación obtenida en la Evaluación Continua.

En cualquier caso, el alumno se puede presentar a cualquiera de los tres niveles, pero solamente a uno de ellos. Solamente se tiene que examinar de los cuestionarios asociados al nivel del que se quiere examinar, y presentar el proyecto asociado a dicho nivel.

10.4. Convocatoria Extraordinaria

La Convocatoria Extraordinaria se realiza en la fecha prevista en el calendario de exámenes. Esta convocatoria está indicada solamente para alumnos que no han aprobado ninguno de los niveles en la Evaluación Continua ni en la Convocatoria Ordinaria.



El alumno se puede presentar a cualquiera de los tres niveles, pero solamente a uno de ellos. Solamente se tiene que examinar de los cuestionarios asociados al nivel del que se quiere examinar, y presentar el proyecto asociado a dicho nivel.

8. Consideraciones finales

