



Universidad
de Alcalá

GUÍA DOCENTE

Sistemas Informáticos

Grado en

Ingeniería en Tecnologías de Telecomunicación (GITT)
Ingeniería en Sistemas de Telecomunicación (GIST)
Ingeniería Telemática (GIT)
Ingeniería Electrónica de Comunicaciones (GIEC)

Universidad de Alcalá

Curso Académico 2022/2023

1^{er} Curso - 1^{er} Cuatrimestre (GITT+GIST+GIT+GIEC)

GUÍA DOCENTE

Nombre de la asignatura:	Sistemas Informáticos
Código:	350003 (GITT+GIST+GIT+GIEC)
Titulación en la que se imparte:	Grado en Ingeniería en Tecnologías de Telecomunicación (GITT) Ingeniería en Sistemas de Telecomunicación (GIST) Ingeniería Telemática (GIT) Ingeniería Electrónica de Comunicaciones (GIEC)
Departamento y Área de Conocimiento:	Automática Arquitectura y Tecnología de Computadores
Carácter:	Básica (GITT+GIST+GIT+GIEC)
Créditos ECTS:	6.0
Curso y cuatrimestre:	1^{er} Curso - 1^{er} Cuatrimestre (GITT+GIST+GIT+GIEC)
Profesorado:	Concha Batanero Ochaíta María del Mar Lendínez Chica Óscar Gutiérrez Molina
Horario de Tutoría:	Consultar entorno de programación docente: https://uah.blackboard.com
Idioma en el que se imparte:	Español

1a. PRESENTACIÓN

Sistemas Informáticos es una asignatura básica que se imparte en el primer cuatrimestre del primer curso de Grado en Ingeniería en Tecnologías de Telecomunicación, Grado en Ingeniería Telemática, Grado en Ingeniería en Sistemas de Telecomunicación y Grado en Ingeniería Electrónica de Comunicaciones.

Esta asignatura no requiere conocimientos previos, e introduce al alumno en los conceptos elementales relacionados con la estructura de los computadores de propósito general actuales. Se describirá y justificará cada uno de los elementos constituyentes desde el punto de vista de la funcionalidad que aporta.

Se introducirá al estudiante en los problemas relativos a la representación de la información en sistemas digitales, desde las formas más elementales hasta algunas estructuras de datos complejas. Además se analizará la forma de representar acciones y procedimientos de forma algorítmica, de forma que puedan ser ejecutados por un microprocesador.

El alumno aprenderá la forma de construir programas utilizando las herramientas básicas proporcionadas por un lenguaje de programación de propósito general.

Esta asignatura supone la base de la asignatura Programación impartida en el segundo semestre. Además aporta conocimientos que serán de utilidad en las siguientes asignaturas correspondientes a los grados en Ingeniería de Telecomunicación: Álgebra Lineal, Electrónica Digital, Diseño Electrónico, Sistemas Electrónicos Digitales, Sistemas Electrónicos Digitales Avanzados, Programación Avanzada, Programación Visual, Sistemas Operativos.

1b. COURSE SUMMARY

Computing Systems is a basic subject imparted in the first term of the first year in several Engineering degrees: Communication Electronics Engineering, Telecommunication Systems Engineering, Engineering on Telecommunication Technologies, and Telematics Engineering.

This subject aims to teach how to build computers programs using the basic tools provided by a general purpose programming language. In order to achieve this final goal elementary concepts related to general purpose computers have to be deal with the subject, therefore every constituent element will be described and justified from the point of view of the provided functionality, likewise, the most relevant cases in information coding, starting with the most elementary forms of data and reaching to medium complexity data structures. Similarly, the methods for coding actions and procedures in algorithmic form will be presented.

The Programming subject, taught in the second term, is based on this subject. Furthermore, Computing Systems provides the knowledge required in several subjects throughout the degree: Álgebra Lineal, Electrónica Digital, Diseño Electrónico, Sistemas Electrónicos Digitales, Sistemas Electrónicos Digitales Avanzados, Programación Avanzada, Programación Visual, Sistemas Operativos Some of them are also offered in English.

In summary, the main concepts gathered in this subject are: digital computer architecture, information coding, elements of the C Language, standard Input/Output, functions, control statements, arrays, structures, arrays of structures, pointers and dynamic memory allocation.

2. COMPETENCIAS

Competencias básicas, generales y transversales.

Esta asignatura contribuye a adquirir las siguientes competencias básicas, generales y transversales definidas en el apartado 3 del Anexo de la Orden CIN/352/2009:

TR2 - Conocimiento de materias básicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías, así como que le dote de una gran versatilidad para adaptarse a nuevas situaciones.

TR3 - Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, y de comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación.

TR5 - Facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.

TR8 - Capacidad de trabajar en un grupo multidisciplinar y en un entorno multilingüe y de comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.

TRU2 - Comunicación oral y escrita.

TRU3 - Capacidad de gestión de la información.

TRU4 - Capacidad de aprendizaje autónomo.

TRU5 - Capacidad para trabajar en equipo.

Competencias de Carácter Profesional

Esta asignatura proporciona la(s) siguiente(s) competencia(s) de carácter profesional definida(s) en el apartado 5 del Anexo de la Orden CIN/352/2009:

CB2 - Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Resultados de aprendizaje

Al terminar con éxito esta asignatura los alumnos serán capaces de:

RA1. Interpretar el funcionamiento de programas sencillos y operaciones en binario.

RA2. Distinguir los tipos básicos de datos y sentencias de control en un programa.

RA3. Identificar los distintos elementos utilizados en la programación estructurada como funciones, arrays y estructuras.

RA4. Programar con punteros y controlar su aplicación en memoria dinámica.

RA5. Codificar programas de forma autónoma en Lenguaje C de dificultad considerable, aplicables directamente a cualquier entorno laboral, científico o tecnológico.

3. CONTENIDOS

Bloques de contenido (se pueden especificar los temas si se considera necesario)	Total de clases, créditos u horas
Fundamentos de ordenadores	2 T + 2 P horas
Fundamentos de programación	14 T + 14 P horas
Tipos estructurados de datos	6 T + 6 P horas
Asignación dinámica de memoria	6 T + 6 P horas

Cronograma

Semana / Sesión	Contenido
01 ^a	<ul style="list-style-type: none"> • Concepto de ordenador. Arquitectura Von Neumann del ordenador digital. Periféricos. Los ordenadores actuales. Programas y lenguajes. Periféricos, Representación de la información. Sistema Operativo. Ejercicios. • Práctica: Introducción al Entorno de Desarrollo Integrado, IDE.
02 ^a	<ul style="list-style-type: none"> • Fases en el desarrollo de un programa. Introducción a la E/S, matrices, sentencias de control y funciones. Ejercicios. • Práctica: Entorno de desarrollo: edición, compilación, ejecución y depuración.
03 ^a	<ul style="list-style-type: none"> • Tipos primitivos de datos. Tipos derivados. Sinónimos de tipos. Literales. Identificadores. Palabras claves. Comentarios. Variables. Constantes simbólicas. Expresiones numéricas. Operadores. Prioridad y orden de evaluación. Conversión entre tipos. Ejercicios. • Práctica: Operaciones con bits, operadores y sentencias de control.
04 ^a	<ul style="list-style-type: none"> • Estructura de un programa. Directrices de inclusión y de sustitución. Declaraciones y definiciones. Sentencias: simple y compuestas. Datos numéricos y cadenas de caracteres. Flujos de entrada y salida estándar. Ejercicios. • Práctica: Operaciones con bits, operadores y sentencias de control.
05 ^a	<ul style="list-style-type: none"> • Funciones: Declaración, definición y llamada a una función. Argumentos por valor y por referencia. Ámbito de las variables. Clases de almacenamiento. Salida con formato. Entrada con formato. Entrada de caracteres y salida de caracteres. Carácter fin de línea y carácter fin de fichero. Validar un dato de entrada. Ejercicios. • Práctica: Operaciones con bits, operadores y sentencias de control.
06 ^a	<ul style="list-style-type: none"> • Sentencias de control secuenciales y repetitivas. Ejercicios. • Práctica: Sentencias de Control.
07 ^a	<ul style="list-style-type: none"> • Matrices. Matrices numéricas unidimensionales. Matrices asociativas. Cadenas de caracteres. Funciones para cadenas de caracteres Tipo y tamaño de una matriz. Matrices multidimensionales. Ejercicios. • Práctica: Arrays y Sentencias de Control.
08 ^a	<ul style="list-style-type: none"> • Matrices de cadenas de caracteres. Copiar matrices. Trabajar con bloques de bytes. Ejercicios. • Práctica: Arrays y Sentencias de Control.

09 ^a	<ul style="list-style-type: none"> • Estructuras. Estructuras y Matrices de estructuras. Ejercicios. • Práctica: Tipos Estructurados.
10 ^a	<ul style="list-style-type: none"> • Creación de punteros. Operadores dirección de y contenido. Operaciones con punteros. Punteros y matrices. Ejercicios. • Práctica: Tipos Estructurados.
11 ^a	<ul style="list-style-type: none"> • Punteros a cadenas de caracteres. Matrices de punteros. Punteros a punteros. Matriz de punteros a cadenas de caracteres. Ordenar matrices de cadenas de caracteres. Ejercicios. • Práctica: Tipos Estructurados.
12 ^a	<ul style="list-style-type: none"> • Asignación dinámica de memoria. Funciones para asignación dinámica de memoria. Ejercicios. • Práctica: Punteros y Memoria Dinámica.
13 ^a	<ul style="list-style-type: none"> • Matrices dinámicas. Punteros a estructuras. Punteros como parámetros en funciones. Ejercicios. • Práctica: Punteros y Memoria Dinámica.
14 ^a	<ul style="list-style-type: none"> • Matrices dinámicas de punteros. Matrices Dinámicas de punteros a punteros. Ejercicios. • Práctica: Punteros y Memoria Dinámica.

4. METODOLOGÍAS DE ENSEÑANZA APRENDIZAJE. ACTIVIDADES FORMATIVAS

4.1. Distribución de créditos

Número de horas presenciales:	58 horas (56 horas de clase presencial +2 horas de evaluación)
Número de horas del trabajo propio del estudiante:	92
Total horas	150

4.2. Estrategias metodológicas, materiales y recursos didácticos

La estrategia metodológica de formación está constituida por los siguientes tipos de actividades:

Sesiones teóricas

Se trata de clases magistrales donde se explica el contenido teórico básico de la asignatura. El objetivo es la adquisición por parte del alumno de la filosofía de programación. Dada la temática eminentemente práctica de la asignatura, las sesiones teóricas irán acompañadas de la exposición de un gran número de ejemplos prácticos

Con el fin de imprimir carácter dinámico a las clases, para la exposición teórica de los contenidos se

utilizarán todos los medios existentes en las aulas: pizarra, proyector de transparencias y ordenador, el cual permite mostrar a los alumnos la resolución y ejecución de programas.

Se propiciará la participación de los alumnos en clase a través del diálogo y consulta colectiva de posibles soluciones de los ejercicios con el fin de promover el aprendizaje colaborativo.

Se solicitará a los alumnos su participación activa en clase mediante la resolución de ejercicios en la pizarra con el fin de poner en común las soluciones adoptadas por los estudiantes.

Para realizar un seguimiento del aprendizaje y estimular el interés por la programación, se podrá solicitar la realización de ejercicios individuales y en grupo y su entrega por escrito al profesor.

Sesiones de laboratorio

El objetivo de estas sesiones es introducir al alumno en el mundo de la programación mediante la realización de programas relacionados con el mundo laboral y profesional. Para ello disponen de ordenadores y del software necesario para la elaboración, compilación, ejecución y depuración de los programas.

Cada alumno puede realizar las prácticas de forma individual, si bien se aconseja formar grupos de un máximo de dos alumnos.

Se realizarán 5 prácticas a lo largo del periodo docente consistentes en la realización de programas, de dificultad creciente, relacionadas con los contenidos y ejemplos explicados en las clases teóricas.

El contenido de las prácticas irá creciendo gradualmente en amplitud y complejidad de forma que los alumnos adquieran de forma progresiva las competencias correspondientes a la asignatura. Inicialmente, las prácticas estarán guiadas mediante pseudocódigo que orientará al alumno en el camino a seguir para la correcta ejecución del programa. A medida que el alumno va adquiriendo competencias se le dotará de libertad para la codificación del programa de acuerdo a sus criterios, adoptando el profesor un papel de tutelaje cuya función será la de orientar en el trabajo.

Durante el proceso de aprendizaje el alumno puede hacer uso tanto de bibliografía como de búsquedas de información en Internet.

Tutorías individuales o grupales

Se establecen horarios de tutorías para facilitar el apoyo individualizado o en grupo de aquellos alumnos que así lo requieran. Además permitirá al profesor un seguimiento de la asimilación de los conceptos explicados por parte de los alumnos. Esto posibilitará la readaptación de las clases de cara a un mayor aprendizaje.

Como soporte y complemento a las actividades mencionadas, los alumnos disponen de una plataforma de aprendizaje donde podrán acceder a diferente información y material de la asignatura. Así podrán tener acceso a:

- Documentación elaborada por los profesores
- Prácticas a realizar
- Exámenes de años anteriores
- Profesorado y horario de tutorías
- Información actualizada de la asignatura mediante foro y noticias
- Fechas y calificaciones de exámenes, etc.

5. EVALUACIÓN: Procedimientos, criterios de evaluación y calificación

Preferentemente se ofrecerá a los alumnos un sistema de evaluación continua que tenga características de evaluación formativa de manera que sirva de realimentación en el proceso de enseñanza-aprendizaje por parte del alumno.

5.1. PROCEDIMIENTOS

La evaluación debe estar inspirada en los criterios de evaluación continua (Normativa de Evaluación de los Aprendizajes, NEA, art 3). No obstante, respetando la normativa de la Universidad de Alcalá se pone a disposición del alumno un proceso alternativo de evaluación final de acuerdo a la Normativa de Evaluación de los Aprendizaje (aprobada en Consejo de Gobierno de 24 de marzo de 2011 y modificada en Consejo de Gobierno de 5 de mayo de 2016) según lo indicado en su Artículo 10, los alumnos tendrán un plazo de quince días desde el inicio del curso para solicitar por escrito al Director de la Escuela Politécnica Superior su intención de acogerse al modelo de evaluación no continua aduciendo las razones que estimen convenientes. La evaluación del proceso de aprendizaje de todos los alumnos que no cursen solicitud al respecto o vean denegada la misma se realizará, por defecto, de acuerdo al modelo de evaluación continua. El estudiante dispone de dos convocatorias para superar la asignatura, una ordinaria y otra extraordinaria.

Para llevar a cabo la evaluación continua se realizarán distintas pruebas a lo largo del periodo docente. Se considerará agotada la convocatoria siempre y cuando el alumno se presente a la última prueba de evaluación continua.

Para optimizar el proceso de enseñanza-aprendizaje, los docentes de la asignatura recomiendan optar por el modelo de evaluación continua.

El modelo de evaluación final se llevará a cabo a través de una única prueba, que coincidirá en fecha, aunque no en forma y contenido, con la última prueba que realizarán los alumnos acogidos al modelo de evaluación continua. La última prueba será, para los alumnos de evaluación continua, una prueba de conjunto escrita y para los de evaluación final, se añadirá a esta prueba una ampliación escrita o mediante la plataforma de aprendizaje Blackboard.

Convocatoria extraordinaria.- los alumnos que no hayan superado la convocatoria ordinaria tendrán a su disposición la convocatoria extraordinaria, consistente en una prueba de conjunto de toda la asignatura.

5.2. EVALUACIÓN

CRITERIOS DE EVALUACIÓN

El objetivo del proceso de evaluación es la comprobación de la adquisición de las competencias de la asignatura. Para ello se definen los siguientes:

- CE1.** El alumno ha adquirido los conceptos técnicos de programación tratados durante el curso y los aplica de forma correcta.
- CE2.** El alumno puede resolver conceptual y correctamente problemas, que, abordando los temas teóricos explicados en clase, sean nuevos y distintos de los resueltos en las clases de ejercicios.
- CE3.** El alumno es capaz de analizar y comprender una problemática o necesidad tecnológica, empresarial, industrial, científica y, en general, de cualquier rama del mundo laboral, proporcionando una solución mediante algoritmos y programación eficiente.
- CE4.** El alumno demuestra el control de la filosofía de programación a través de la utilización y buen uso de estructuras de datos, sentencias de control, punteros y memoria dinámica.
- CE5.** El alumno muestra capacidad para entender, codificar y modificar programas de dificultad considerable formados por múltiples funciones y ficheros.

INSTRUMENTOS DE EVALUACIÓN

En esta sección se presentan los instrumentos de evaluación aplicados a los criterios de Evaluación. Todos ellos se realizarán a través de la plataforma de aprendizaje Blackboard que alberga la página

Web.

1. Prueba de Evaluación Intermedia (PEI): se realizará una prueba sobre conocimientos básicos de la asignatura que englobará la mitad del temario del curso. La prueba se realizará en el laboratorio y estará compuesta por preguntas de opción múltiple y un ejercicio de programación.
2. Pruebas de laboratorio (PLn): habrá un máximo de 4 pruebas donde todas las cuestiones estarán basadas en los ejercicios realizados durante las sesiones del laboratorio correspondientes a las prácticas de la asignatura.
3. Examen Final (PEF): se realizará una prueba de conjunto consistente en preguntas de tipo test y programación de ejercicios propuestos. La PEF tendrá lugar el día propuesto por la Escuela Politécnica en los laboratorios del Departamento o en las aulas.

CRITERIOS DE CALIFICACIÓN

Esta sección cuantifica los criterios de evaluación para la superación de la asignatura.

Convocatoria Ordinaria, Evaluación continua

En la convocatoria ordinaria – evaluación continua la relación entre los criterios, instrumentos y calificación es la siguiente.

Competencia		Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
TRU4	CB2, TR2, TR3, TR8, TRU2	RA1, RA2, RA3	CE1-CE5	PEI	40%
	TR3, TR5, TR8, TRU2, TRU3, TRU5	RA1–RA5	CE1-CE5	PLn	20%
	CB2, TRU2, TRU3, TRU5, TR2, TR3, TR5, TR8	RA1–RA5	CE1-CE5	PEF	40%(100%)

En caso de que la puntuación obtenida en la PEF de lugar a una mayor calificación global de la asignatura que la obtenida mediante la aplicación de la evaluación continua, se asignará el porcentaje del 100% sobre la nota total de la asignatura. En caso contrario se asignará el 40%. En caso de no asistir a la PEF la calificación global de la asignatura será No Presentado (NP)

Asistencia al laboratorio: se considera requisito imprescindible la asistencia a un mínimo del 80% de las sesiones de laboratorio. Si un alumno presenta una tasa de ausencia no justificada superior a 3 sesiones de laboratorio, obtendrá una calificación de no presentado. En las sesiones de laboratorio, los alumnos, mediante la supervisión y seguimiento del profesor, realizarán en el ordenador los programas propuestos en las prácticas atendiendo a la resolución de los errores de compilación y ejecución hasta conseguir la correcta ejecución de los programas.

El hecho de aprobar cualquiera de las pruebas de evaluación continua no supondrá en ningún caso la liberación de la materia para posteriores convocatorias.

Convocatoria Ordinaria, Evaluación Final

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CB2, TRU2, TRU3, TRU4, TRU5, TR2, TR3, TR5, TR8	RA1-RA5	CE1-CE5	PEF	100%

Convocatoria extraordinaria

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CB2, TRU2, TRU3, TRU4, TRU5, TR2, TR3, TR5, TR8	RA1-RA5	CE1-CE5	PEF	100%

Esta prueba se realizará en los laboratorios del Departamento o en las aulas en la fecha propuesta por la Escuela Politécnica para los exámenes extraordinarios y estará compuesta por preguntas tipo test y ejercicios de programación.

6. BIBLIOGRAFÍA

6.1. Bibliografía básica

- Curso de programación con C/C++. Ed. RA-MA. Fco. Javier Ceballos.

6.2. Bibliografía complementaria

- El lenguaje de programación C. Ed. Prentice Hall. Kernighan y D. M. Ritchie.
- Como programar en C/C++. Ed. Prentice Hall. H.M.Deitel & P.J.Deitel.
- Introducción a la Informática. McGraw Hill. Prieto, Lloris, Torres.

NOTA INFORMATIVA

La Universidad de Alcalá garantiza a sus estudiantes que, si por exigencias sanitarias las autoridades competentes impidieran la presencialidad total o parcial de la actividad docente, los planes docentes alcanzarían sus objetivos a través de una metodología de enseñanza-aprendizaje y evaluación en formato online, que retornaría a la modalidad presencial en cuanto cesaran dichos impedimentos.