

**Proyecto/Guía docente de la asignatura**

Asignatura	Informática Industrial		
Materia	Ingeniería de sistemas		
Módulo	Tecnología específica: Electrónica Industrial y Automática		
Titulación	Grado Ingeniería Electrónica Industrial y Automática		
Plan	452	Código	42380
Periodo de impartición	Primer cuatrimestre	Tipo/Carácter	obligatoria
Nivel/Ciclo	Grado	Curso	3º
Créditos ECTS	6		
Lengua en que se imparte	Castellano		
Profesor/es responsable/s	Rogelio Mazaeda Echevarría Gregorio Ismael Sainz Palmero		
Datos de contacto (E-mail, teléfono...)	Escuela Ingenierías Industriales Dpto. Ingeniería de Sistemas y Automática Gregorio Sainz Palmero Sede Mergelina, Despacho 2105 Tel 983 423357 gregorioismael.sainz@uva.es Rogelio Mazaeda Echevarría Sede Mergelina, Despacho 2109 Tlf 983423162 rogelio@cta.uva.es		
Departamento	Ingeniería de Sistemas y Automática		
Fecha de Revisión del Comité de Título	18/06/2025		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Resulta inconcebible la industria contemporánea sin considerar el papel central que juegan los ordenadores y la informática en la misma. Se puede decir que la importancia de la informática en la industria está fundamentalmente ligada al concepto de la automatización en su sentido más amplio de sustitución, total o parcial, de la labor humana en la realización de los procesos involucrados. Pero si se analiza más en detalle, se advierte la multiplicidad de las tareas concretas en las cuales se utiliza el ordenador y que van desde la administración de los procesos de negocio, líneas de suministros, planificación de los recursos y relaciones con los clientes y proveedores en el plano de la gerencia de la empresa, hasta su uso para el control directo de los procesos productivos.

En este ámbito del control de procesos se utilizan dispositivos programables especializados como los PLCs potenciados por el despliegue creciente de redes de campo que cuentan con la posibilidad de realizar procesamiento distribuido. Existe otro ángulo de enfoque de la informática industrial y que se refiere a su uso en la propia conformación de muchos de los productos fabricados, que pueden contener procesadores empotrados utilizados para el control de su funcionamiento. De manera que, los ordenadores y la informática forman parte integral de muchos productos industriales, que son diseñados utilizando software especializado y cuya producción en serie ocurre en fábricas controladas, a diferentes niveles, por computadores digitales.

Queda por tanto claro que los ordenadores y la informática constituyen el elemento técnico integrador de la industria moderna y ocupa la situación central de privilegio que en su momento ocuparon la mecánica a finales del siglo XVIII en los inicios de la revolución industrial y la electricidad a finales del XIX y durante la primera mitad del siglo XX. Esta función de integración sería, por supuesto, imposible sin la existencia de las redes de comunicaciones informáticas.

El preparar un programa de estudios para abordar los contenidos de la informática industrial es un reto pues es preciso condensar un objeto de estudio extremadamente amplio, diverso y dinámico. Se trata, de incidir sobre los fundamentos esenciales y más perdurables de una disciplina que precisamente se caracteriza por la gran volatilidad de las soluciones técnicas concretas. Sin eludir el imprescindible grado de actualidad, es importante encontrar un relato común que ayude al alumno a situar las continuas variantes y que le oriente en la puesta al día constante que necesitará realizar durante toda su vida profesional.

Un estudio de la informática en la industria tendría que hacer referencia a temas que son objeto de estudio fundamental del área, como son el control discreto por ordenador. A nivel introductorio se deberían explicar temas concernientes a la elección del periodo de muestreo, a la necesidad de contar con filtros anti-alias y a las implicaciones del tamaño finito de la palabra del ordenador. El alumno debería ser también capaz de entender y programar un controlador digital. Estos temas no se abordarán tampoco en *Informática Industrial* ya que se imparten en la asignatura obligatoria *Diseño de sistemas de control* en el sexto cuatrimestre.

El núcleo de la asignatura será el estudio de la programación concurrente y los conceptos de sistemas de tiempo real. La característica distintiva de la informática en la industria es el hecho de que tiene que lidiar con el *mundo real*. Ya se trate de la implantación de un control digital directo de un proceso industrial, del controlador empotrado de un equipo, etc.

todos tienen en común el hecho de que no sólo deben entregar el resultado correcto, sino que deben de hacerlo cumpliendo con determinados plazos que son expresados en término del tiempo físico real. Los plazos pueden ser de órdenes de magnitud muy disímiles, en el orden de los milisegundos a los minutos para los primeros ejemplos, o de las horas o días para el caso del último, y las consecuencias de incumplirlos pueden ser más o menos graves o incluso catastróficas si se tratare de un sistema de control de vuelo de un avión; pero en todos los casos se comparte la mencionada similitud esencial.

Los programas informáticos convencionales siguen un paradigma secuencial: las sentencias se ejecutan el orden estricto de su aparición y esta estructura por defecto sólo es alterada ligeramente mediante las sentencias de control de flujo para crear bucles o condicionales. Pero en el mundo físico, con el que se debe interactuar en la Informática Industrial, las cosas ocurren, naturalmente, de forma simultánea. Es evidente que se podría utilizar un programa secuencial para interactuar con los eventos simultáneos reales, pero resulta conceptualmente más apropiado utilizar el paradigma de la programación concurrente. En este caso se ejecutan simultáneamente varios procesos o hilos que han sido diseñados sin asumir ningún orden específico de ejecución entre ellos. Se han diseñado simplemente para que cada uno esté dedicado a la atención de determinado aspecto concreto de los procesos a controlar. Es cierto que en un computador con un solo procesador no existe simultaneidad real pero el modelo concurrente permite abordar el paralelismo inherente al mundo real. Es además claramente más ventajoso en los cada vez más habituales sistemas multiprocesadores. No obstante, la programación concurrente tiene una mayor complejidad, sobre todo en el caso en que las diferentes tareas tengan que intercambiar información o coordinar sus funciones de alguna manera.

La **seguridad y fiabilidad de las aplicaciones informáticas** desplegadas en la industria, también se tendrán en cuenta en la asignatura. El fallo de las mismas, a la hora de realizar su cometido, puede tener consecuencias personales o económicas graves, o ser incluso catastróficas, dependiendo de la industria y aplicación concreta. La seguridad de las soluciones de automatización, se mueven progresivamente desde los tradicionales enclavamientos eléctricos fijos a soluciones basadas en equipos programables. El diseño, la implementación y la verificación de la seguridad de las aplicaciones informáticas industriales tiene, un papel cada vez más relevante y deben ser abordadas en una asignatura de estas características.

Se ha intentado que los contenidos que se imparten en la asignatura tengan un grado aceptable de perdurabilidad. Para ello, en primer lugar, se ha acudido a conceptos asentados como los mencionados: modelos concurrentes de programación, conceptos de sistemas de tiempo real y la explicación de las diferentes estrategias a la hora de programar los diferentes ejecutivos cíclicos estáticos y dinámicos, analizando el grado de aprovechamiento del tiempo de procesamiento, entre otros conceptos de valor estable. La asignatura, sin embargo, no puede eludir el aspecto práctico. Por ello, para dotar a la asignatura de la actualidad necesaria, sin sucumbir a la utilización de productos y aplicaciones que se demuestren de importancia efímera o demasiado ligadas a soluciones particulares, la estrategia ha consistido en apelar a soluciones basadas en estándares industriales *de facto* o sancionados por organismos internacionales de reconocido impacto en el mundo industrial. Así, por ejemplo, la ilustración práctica de los conceptos de programación concurrente se realizará en **lenguaje C++11**. El lenguaje C ha sido estudiado en la asignatura básica de *Fundamentos de Informática* durante el primer año del curso. Las soluciones en C++ permiten la programación de soluciones de informática industrial brindando un gran nivel de generalidad.

1.2 Relación con otras materias

Como se ha mencionado en el apartado anterior, la asignatura de *Informática Industrial* tiene una relación muy estrecha con otras muchas asignaturas de la titulación de Grado en Electrónica Industrial y Automática.

En particular con las asignaturas pertenecientes a la materia de Automática, tiene relaciones evidentes con las asignaturas de *Automatización Industrial* que se imparte paralelamente en el mismo cuatrimestre (C5) y con *Control y Comunicaciones Industriales* de cuarto del curso, primer cuatrimestre (C7), ambas de carácter obligatorio. Con la primera de ellas, se podrá crear un diálogo interesante al tener un objeto de estudio con área con área comunes muy amplias, pero que se verán desde ópticas y con énfasis diferentes.

Con respecto a la segunda de las asignaturas mencionadas, el tratamiento sobre los temas que en común tiene con ella *Informática Industrial*, puede verse como una introducción a los temas de redes informáticas en la industria, los que serán tratadas con más profundidad en *Control y Comunicaciones Industriales* posteriormente.

Existen relaciones también importantes, con las asignaturas de *Control de Procesos* (OP) de 4º año segundo cuatrimestre (C8) perteneciente a la mencionada materia de automática.

En la propia materia de Ingeniería de Sistemas, existen vínculos con *Modelado y Simulación de Sistemas* (OB de C6).

Las asignaturas de *Sistemas Robotizados* (OB, C7) y *Taller de Robótica Industrial* (OP, C8) tienen como objetos de estudio temas, que en rigor tienen componentes de informática industrial, como sería los programas de control de los robots industriales, pero salvo alguna referencia general, estos temas no serán abordados en la asignatura que nos ocupa.

Evidentemente, existen vínculos muy estrechos con las asignaturas básicas y obligatorias que son prerequisites de *Informática Industrial* y que se describen a continuación.

1.3 Prerrequisitos

En la asignatura Informática Industrial se asumen como conocimientos previos de los alumnos los impartidos en las asignaturas:

- *Fundamentos de Informática*: asignatura básica de primer curso que presenta los principios de programación de ordenadores y de Sistemas Operativos.
- *Fundamentos de Automática*: asignatura obligatoria perteneciente al módulo común a la rama industrial que se imparte en el segundo curso de la titulación y que estudia los conceptos básicos de sistemas dinámicos, realimentación, estabilidad y diseño de controladores.
- *Sistemas de Producción y Fabricación*: Asignatura (OB) que pertenece al módulo común a la rama industrial brinda una introducción útil a determinados conceptos clave de la automatización industrial.

2. Competencias

La asignatura debe ayudar a desarrollar las siguientes competencias generales y específicas.

2.1 Generales

La asignatura busca contribuir a desarrollar las siguientes capacidades de carácter general:

- CG1: Capacidad de análisis y síntesis.
- CG2: Capacidad de organización y planificación del tiempo.
- 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.
- CG15: Capacidad para el manejo de especificaciones técnicas y la elaboración de informes técnicos.

2.2 Específicas

En particular, se tratará de desarrollar la competencia específica:

- CE28: Conocimiento aplicado de informática industrial y comunicaciones.

3. Objetivos

El propósito último de la asignatura, tal y como se plantea en el descriptor aprobado por la agencia de acreditación, es el de contribuir a lograr que el estudiante sea capaz de:

- Analizar, desarrollar e implementar proyectos informáticos que incluyan la integración de software en el ámbito de la automatización utilizando equipos específicos y técnicas de programación orientada a objeto.
- Realizar y evaluar proyectos de diseño, especificación, implementación, validación y actualización de sistemas que incluyan sistemas de cómputo, sistemas embebidos y redes de ordenadores.
- Desarrollar proyectos de explotación y mantenimiento de aplicaciones informáticas que incluyan sistemas industriales en entornos distribuidos.

De forma más concreta, los objetivos evaluables perseguidos, son los siguientes:

- Comprender las diferentes arquitecturas y formas de utilización del ordenador y la informática en la industria moderna. Identificar las características fundamentales del funcionamiento y la configuración para los diferentes tipos de aplicaciones industriales utilizadas en cada nivel de la empresa.
- Conocer los principios y modelos fundamentales de la programación concurrente. Ser capaz de implementar programas concurrentes utilizando el lenguaje de programación orientado a objeto C++.
- Conocer los conceptos fundamentales de los sistemas operativos en tiempo real. Entender cómo se aplican los conceptos de programación concurrente y tiempo real en controladores industriales.
- Conocer las implicaciones de la seguridad y la tolerancia a fallos aplicada a la informática industrial.

4. Contenidos y/o bloques temáticos

Los contenidos de la asignatura se estructuran en un único bloque de 8 temas de la forma que a continuación se especifica.

Bloque 1:

Carga de trabajo en créditos ECTS: 6 ECTS

a. Contextualización y justificación

El núcleo de la asignatura será el estudio de programación concurrente y de los conceptos de lenguajes de programación y sistemas de tiempos real. La característica distintiva de la informática en la industria es el hecho de que tiene que interactuar con el “mundo real”. Todos los sistemas industriales tienen en común el hecho de que no sólo deben entregar el resultado correcto, sino que deben de hacerlo cumpliendo con determinados plazos que son expresados en término del tiempo físico real.

Los programas informáticos convencionales siguen un paradigma secuencial: las sentencias se ejecutan el orden estricto de su aparición y esta estructura por defecto sólo es alterada ligeramente mediante las estructuras conocidas de control de flujo para crear bucles o condicionales. Pero en el mundo físico, con el que se debe interactuar en la Informática Industrial, las cosas ocurren, naturalmente, de forma simultánea. Es evidente que se podría utilizar un programa secuencial para interactuar con los eventos simultáneos reales, pero resulta conceptualmente más apropiado utilizar el paradigma de la programación concurrente, donde se tienen varios procesos o hilos secuenciales de ejecución, en cuyo diseño no se asume ningún orden específico de ejecución entre ellos, cada uno dedicado a la atención de determinado aspecto independiente de los procesos a controlar. Por otro lado, aunque en un computador con un solo procesador no existe simultaneidad real, el modelo concurrente es claramente más eficiente en los sistemas multiprocesadores que son cada vez cada vez más habituales. La programación concurrente no obstante tiene sus desventajas, asociadas a una mayor complejidad, sobre todo en el caso en que las diferentes tareas tengan que intercambiar información o coordinar sus funciones de alguna manera.

El bloque comienza la introducción los conceptos de programación orientada a objeto en C++ y la presentación de las siguientes estructuras de datos: pilas, colas y listas que se programarán en el laboratorio empleando la biblioteca STL.

Los conceptos de programación concurrente y en tiempo real son fundamentales en la informática industrial, pero la forma más habitual de aplicarlos no es el realizar la programación a nivel de lenguajes de propósito general sino el hacerlo a un nivel superior de abstracción, programando dispositivos tales como los autómatas programables y/o desplegando y configurando sistemas de Control Distribuido (DCS) y sistemas de supervisión tipo SCADA y que se elegirán de acuerdo al nivel de que se trate y a las características de la aplicación a realizar.

Por otro lado, los dispositivos programables industriales cada vez con más frecuencia asumen funciones cuyo fallo puede, no sólo conducir a importantes pérdidas económicas sino también a graves daños personales. Es importante, en consecuencia, recurrir a métodos contrastados para verificar el comportamiento de los programas que ejecutan estos dispositivos en relación con los elementos físicos con los que interactúan. Existe un importante intento de estandarización sobre la así llamada seguridad funcional que está representado por la IEC 61508 y sus normas derivadas, que definen y utilizan conceptos como los de riesgo, riesgo tolerable, función instrumentada de seguridad (SIF), nivel integrado de seguridad (SIL).

b. Objetivos de aprendizaje

- Conocer la programación orientada a objeto. Ser capaz de programar estructuras de datos avanzadas en C++.
- Conocer los principios y modelos fundamentales de la programación concurrente. Ser capaz de implementar programas concurrentes utilizando el lenguaje C++ y el estándar POSIX.
- Comprender las diferentes arquitecturas y formas de utilización del ordenador y la informática en los diferentes niveles de la empresa.
- Conocer y comprender los sistemas OPC como estándar de interoperabilidad.
- Ser consciente de las consecuencias que puede tener un fallo en un sistema informático y conocer los mecanismos de prevención y tolerancia a fallos.

c. Contenidos

Te ma s	Descripción	Horas (T)	Horas (P)
1	Programación C++ y OOP - Introducción Informática Industrial, Industria 4.0 y Programación orientada objeto - C++: - Sintaxis y estructura C++ - Plantillas: STL. - Conceptos OOP: Objetos y Clases en C++	13	26
2	Programación concurrente. - Concurrencia y Señales - Relación entre procesos. Procesos cooperativos - El problema del productor-consumidor. MQTT - Programación de hilos en C++	4	7
3	OPC-UA: Interoperabilidad. - Arquitectura OPC-UA - Programación Cliente-servidor OPC-UA	3	5
4	Ciberseguridad - Conceptos básicos de Ciberseguridad Industrial - Estándares Normativa	2	0

d. Métodos docentes

Los métodos docentes a utilizar serán los mostrados en la siguiente tabla.

MÉTODOS DOCENTES	OBSERVACIONES
Método expositivo /lección magistral.	Se utilizará fundamentalmente en las clases de aula de teoría en las que se introducen nuevos conceptos.
Resolución de ejercicios y problemas.	Se pondrán en práctica las clases de teoría como forma de motivar conceptos nuevos, pero sobre todo formarán la base del trabajo en el aula de ordenadores. Algunos problemas se plantearán de forma poco estructurada y realista, de forma que tengan aproximarse a ellos utilizando e integrando recursos aprendidos a lo largo del curso.
Aprendizaje orientado a proyectos.	El peso fundamental de este método se concentrará en la realización de dos proyectos evaluables. Su realización es fundamentalmente no presencial. No obstante, el proyecto

	será presentado y se realizará la orientación en las clases de prácticas. También resultarán de mucha ayuda en este ámbito las tutorías.
--	--

En lo que sigue, se detallará la metodología y técnicas docentes a seguir, en los diferentes encargos docentes. Se destacarán los medios técnicos y recursos docentes a utilizar en cada caso.

d.1 Actividades presenciales

Entre las actividades presenciales se tienen: las clases de teoría en el aula, las clases prácticas frente al ordenador y las tutorías.

d.1.1. Clases de Teoría en aula

Se impartirán los elementos teóricos de la asignatura siguiendo el modelo de **lección magistral** y, en la medida en que sea posible, el de **lección magistral participativa**. La asignatura presenta retos importantes debido a su diversidad. Esto requiere realizar un ejercicio de contención, respetando la máxima de que *no todo lo que puede ser enseñando, debe enseñarse*. Por ejemplo, del conjunto de múltiples funciones que pone el estándar POSIX para el manejo de la concurrencia y comunicación entre tareas, se presentarán un subconjunto mínimo suficiente, ocultando detalles que más que aclaran, dificultan la comprensión del alumno. El alumno interesado, siempre podrá profundizar en el trabajo autónomo, guiándose de los textos recomendados en la bibliografía o utilizando el material diverso, se pone a su disposición en **Moodle**.

La asignatura tiene un componente decisivo de programación. Y el estudio de la programación de ordenadores tiene características particulares en el sentido de que es muy difícil estudiar los conceptos de forma aislada. Cada nuevo elemento se integrará en el cuerpo de lo dado hasta ese momento y esta integración se intentará hacer estimulando la participación de los alumnos mediante preguntas concretas y utilizando ejemplos simples a resolver en la pizarra.

Para impartir las clases teóricas se utilizarán **medios técnicos** como **PowerPoint**, animaciones representando, por ejemplo, la traza o ejecución paso a paso de los programas. El uso de recursos como el **PowerPoint** es útil para la presentación contenido nuevo, pero se tiene especial cuidado de combinar su uso, con una frecuente utilización de la pizarra, dando al alumno el tiempo requerido para asimilar los contenidos. Las transparencias están disponibles, con suficiente antelación al desarrollo de la clase presencial, en la página de **Moodle**. En esta última también se coloca material extra cuya necesidad surja de la interacción con el estudiante en clases y de la necesidad de nuevas aclaraciones que dicho intercambio sugiera, además del material avanzado ya mencionado y que puede incluir textos explicativos y manuales, fragmento de códigos de programación, enlaces a páginas web de interés.

d.1.2 Clases de Prácticas en Sala de Ordenadores

Las clases prácticas frente al ordenador constituyen un elemento esencial del aprendizaje de una asignatura como Informática Industrial. Se persigue un tipo de aprendizaje basado en la experiencia, al poner en práctica los contenidos teóricos impartidos en el aula.

Las actividades que tienen como fin el adiestramiento con las herramientas (Linux, lenguaje C++ y compilación) se pueden desarrollar de forma individual. Sin embargo, los trabajos obligatorios se podrán realizar individualmente o por parejas que se configurarán a voluntad de los propios estudiantes.

Las prácticas se realizarán en aula de ordenadores. Las clases prácticas son de dos horas de duración. Las prácticas se realizarán empleando el IDE de *Code:blocks*, un entorno de trabajo que los estudiantes conocen desde Fundamentos de Informática en el primer año.

El modelo seguido es el de aprendizaje mediante experiencias introduciendo en algunas ocasiones algunos elementos de métodos del caso, aunque con matices diferenciales en cada una de una de las dos horas. En la primera hora, por ejemplo, el papel del profesor es más destacado: se presentan casos realistas pero sencillos, que hagan hincapié en la utilización de los nuevos conceptos que han sido impartidos durante la semana, pero también buscando la integración necesaria con los contenidos previos. Aquí el profesor guiará la discusión hacia la comprensión de aquellos elementos prácticos sobre los que no ha habido la oportunidad de profundizar en las clases teóricas. Se atenderán las dudas individuales y colectivas de los alumnos, haciendo énfasis en la adquisición de habilidades que permitan interpretar, por ejemplo, la forma en que reportan los errores sintácticos y las mejores prácticas de depuración de los errores semánticos durante la ejecución de los programas elaborados.

En la segunda hora, el énfasis se pone en el trabajo individual del alumno, aunque también con puntuales intervenciones del profesor. Se les presenta ahora un problema de mayor entidad o que no esté demasiado bien estructurado en el cual el esfuerzo de integración de todos los elementos dados hasta ese punto y la necesaria creatividad del alumno, se hacen más exigentes. Los problemas pueden resolverse en el plazo de una hora, aunque no es extraño que requieran por parte del alumno el continuar el esfuerzo en la etapa de estudio individual y colectivo posterior, de carácter no presencial. Los programas elaborados pueden ser enviados al profesor a través de **Moodle** para solventar las dudas que pudieran surgir o para ser posteriormente discutidos en tutorías.

Los alumnos podrán disponer con tiempo suficiente previo a la práctica del guión de la misma, de forma que puedan prepararla con antelación.

Los medios técnicos para la docencia que se utilizarán, además evidentemente de los ordenadores, compiladores y herramientas objeto del estudio, serán:

- **Campus Virtual (Moodle)** para alojar los guiones de las prácticas y material explicativo. También tendrá habilitada la posibilidad, por parte del alumno, de cargar los ficheros de la práctica que haya podido terminar de forma autónoma y los informes de la misma. El foro de Moodle, ofrece un complemento importante para presentar dudas respecto a la práctica.
- **Powerpoint:** Se utilizará para resaltar algún punto importante de interés general, que complemente lo que se está tratado en la práctica.

d.1.3 Tutorías

El profesor estará accesible a las consultas voluntarias de los alumnos, ya sean estas individuales o de grupo. Los horarios de tutoría y atención al alumnado pueden consultarse en la página web de la Uva.

No obstante, es recomendable acordar previamente una cita con el profesor. Los profesores atenderán gustosamente consultas por **correo electrónico**.

d.1.4 Actividades no presenciales

En el EEES se establece que por cada crédito ECTS el alumno debe dedicar aproximadamente 25 horas de estudio, incluyendo actividades presenciales y no presenciales.

De manera que, en correspondencia con las 60 horas de actividad presencial descritas en el epígrafe anterior, se corresponde como mínimo 90 horas de estudio autónomo no presencial, según se recoge en la tabla 4.

Entre las actividades no presenciales a realizar, se tiene:

- Estudio con vista a los exámenes.
- Preparación previa y análisis posterior a las prácticas y a las clases teóricas.
- Realización de trabajos evaluables.
- Preparación previa y análisis posterior a las tutorías.

Para facilitar las actividades no presenciales se procurará que los requisitos técnicos del puesto de trabajo y del software permitan la ejecución de los trabajos desde el domicilio del estudiante y con un equipo informático convencional. En este sentido se favorecerá el empleo de software libre y contenidos abiertos.

Un medio técnico de apoyo fundamental, para el estudio no presencial, lo ofrece la plataforma **Moodle**, al permitir el vínculo a distancia entre profesor y alumnos. Este vínculo no debe resultar en ningún caso perjudicial para la necesaria independencia del estudiante; pero permite mantener el necesario diálogo, utilizando recursos como los foros de discusión.

e. Plan de trabajo

Los conceptos y principios discutidos en clases de teoría e impartidos según la programación discutida previamente, serán apoyados por las clases prácticas y las tutorías como actividades presenciales y por el estudio autónomo individual y colectivo.

Las prácticas se realizarán frente al ordenador con objeto de ayudar a fijar los conceptos teóricos: programación C++ de estructuras de datos y programación C++ de comunicación entre procesos. En las prácticas de programación se irán explicando ejemplos similares, de carácter práctico, a través de los diferentes niveles de abstracción representados por la programación en lenguaje C++.

f. Evaluación

Veáse apartado 7.

g Material docente

Estarán disponibles en el Campus Virtual:

- Transparencias de la asignatura
- Guiones de prácticas
- Descripción de las tareas
- Casos de estudio
- Videos en forma de píldoras de conocimiento

g.1 Bibliografía básica

No existe un libro de texto que cubra todos los temas tratados de forma unitaria. Tampoco la cantidad y calidad de la bibliografía consultable para cada tema es uniforme. Sobre los temas más asentados y estables, como concurrencia, OPC-UA, etc. se accede fácilmente a muchos web sites con información actualizada con diferente grado de profundidad y de detalle en el tratamiento de los temas. La documentación del estándar está pensada para transmitir información, sin ambigüedad a los especialistas, pero tiene escasos valores didácticos.

Como bibliografía básica en soporte impreso incluiríamos únicamente *Programming: Principles and Practice Using C++* 2nd Edition. Bjarne Stroustrup. May 2014. Addison-Wesley Professional ISBN: 978-0321992789. Como bibliografía online citaremos el Tutorial de www.learncpp.com por ajustarse perfectamente a los contenidos de la asignatura.

- *Programming: Principles and Practice Using C++* 2nd Edition. Bjarne Stroustrup. May 2014. Addison-Wesley Professional ISBN: 978-0321992789
- *Cómo Programar C++* - 9ª Edición, Paul Deitel, Pearson Publisher, 2015, ISBN: 978-6073227391
- Tutorials to help you master C++ and object oriented programming <https://www.learncpp.com/>
- Tutorial UML: [Tutorial UML: https://www.tutorialspoint.com/uml/index.htm](https://www.tutorialspoint.com/uml/index.htm)
- OPC-UA: <https://opcfoundation.org/>

g.2 Bibliografía complementaria

- *Effective STL* 50 Specific Ways to Improve Your Use of the Standard Template Library (Scott Meyers) Addison-Wesley Professional; junio de 2001. ISBN-13: 978-0201749625
- *C++ Concurrency in Action. Practical Multithreading.* Anthony Williams. February 2012 ISBN 9781933988771 528 pages Manning Publications.
- Matlab OPC-UA Toolbox: <https://es.mathworks.com/help/opc/>
- Matlab y OOP: <https://es.mathworks.com/company/newsletters/articles/introduction-to-object-oriented-programming-in-matlab.html>
- Tanenbaum, A. *Sistemas distribuidos: principios y paradigmas.* Pearson, 2008.
- Smith, J.B. *Functional Safety: a straightforward guide to IEC 61508 and related standards.* ISA, 2004.

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

UMLetino : <https://www.tutorialspoint.com/uml/index.htm>

h. Recursos necesarios

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

Plataforma educativa para publicar material docente y para hacer más efectiva la comunicación con el alumno.

Acceso al material bibliográfico recomendado.

Distribución Anaconda, con intérprete Python 3, entorno web Jupyter Notebooks y entorno de desarrollo Spyder. Compilador de C/C++ (gcc) y entorno de desarrollo Code::Blocks.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
60 ECTS (22 T + 38 L)	Semanas 1 -14

5. Métodos docentes y principios metodológicos

Ver apartado 4.d página 12.

6. Tabla de dedicación del estudiante a la asignatura

- (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.

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	22	Estudio y trabajo autónomo individual	90
Clases prácticas de aula (A)			
Laboratorios (L)	38		
Prácticas externas, clínicas o de campo			
Seminarios (S)			
Tutorías grupales (TG)			
Evaluación			
Total presencial	60	Total no presencial	90

7. Sistema y características de la evaluación

INSTRUMENTO/ PROCEDIMIENTO	PESO EN NOTA FINAL	OBSERVACIONES
2 Pruebas de laboratorio a realizar en el cuatrimestre.	45%	Evaluación de dos pruebas a realizar en el laboratorio en fecha que se anunciará al comenzar el curso: • 1ª Prueba 20% • 2ª Prueba 25% Se tendrá en cuenta: • Cumplimiento de las especificaciones y metodología: 50% de la nota. • Implementación, funcionamiento y desempeño de la práctica: 50% de la nota. Se requiere un mínimo de 4 sobre 10 en esta prueba para la nota final.
Evaluación de actividad y desempeño en clase	10%	Evaluación de actividad, desempeño y participación en sesiones teóricas. Cada dos semanas se realizará una prueba teórica de 10 minutos sobre los contenidos teóricos desarrollados en ese periodo de clases.
Prueba final	45%	Consistirá en: • Cuestiones. • Problemas/Programación. Se requiere un mínimo de 4 sobre 10 en esta prueba para la nota final.

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**

La nota final de la asignatura es la suma de la nota de la *prueba final del cuatrimestre*, la nota obtenida en las pruebas de laboratorio y la evaluación continua en clase. Es imprescindible haber alcanzado el mínimo marcado en la *prueba final y en las pruebas de laboratorio* para poder aprobar la asignatura.

Caso de no alcanzar el mínimo en algunas de las pruebas del cuatrimestre, la calificación numérica del alumno (suspense) será la correspondiente a dicha parte.

- **Convocatoria extraordinaria:** Prueba de programación en laboratorio y de cuestiones/problemas de programación. Criterios de calificación idénticos a la convocatoria ordinaria.



- **Convocatoria Fin de Carrera:** Idéntica a la convocatoria extraordinaria.

Todos los elementos evaluables deberán ser de autoría exclusiva del estudiante. La utilización de herramientas de Inteligencia Artificial (IA) u otros recursos que no constituyan producción original del alumno conllevará una calificación de 0 sobre 10 en la convocatoria correspondiente. El profesor podrá proponer el uso de herramientas de IA en casos y tareas específicas, pero el estudiante siempre tendrá que especificar la herramienta de IA utilizada y el uso que se ha hecho de la misma.

8. Consideraciones finales

La docencia será presencial, pero por razones organizativas del Centro y de la UVa, algunas actividades podrán impartirse de forma remota.

Las consultas realizadas mediante correo electrónico se responderán lo antes posible, esperando no pasar de 48-72 horas en horario laboral.