

SISTEMAS EN TIEMPO REAL (I. ELECTRÓNICA)

Curso 2012/2013

(Código:6802306-)

1. PRESENTACIÓN DE LA ASIGNATURA

La programación de sistemas en tiempo real es una disciplina compleja ya que el desarrollo de código para este tipo de sistemas requiere una especificación de los "límites" de los sistemas en cuanto a temporización de las tareas y las facilidades de comunicación y sincronización entre procesos del sistema. Adicionalmente es muy relevante la consideración sobre la restricción de ejecución de las tareas y sus prioridades, es decir, la capacidad de decidir que tareas deben ejecutarse, ya que la falta de tiempo para estas puede provocar estados de emergencia en el propio sistema (imagine que falla el software de navegación de un avión). Para resolver estos problemas se introducen conceptos como fiabilidad y tolerancia a fallos, que permiten diseñar sistemas en tiempo real fiables y no propensos a fallos en el propio software. La asignatura pretende mostrar dichos mecanismos y las diferentes estructuras software que permiten la implementación de estos objetivos básicos, en cuanto a: temporización estricta de la ejecución de las tareas de proceso, facilidad de comunicación sin bloqueo de procesos y estabilidad del software.

2. CONTEXTUALIZACIÓN EN EL PLAN DE ESTUDIOS

La programación de sistemas en tiempo real es una disciplina específica de cierto tipo de sistemas que tienen una serie de restricciones temporales en cuanto a la ejecución planificada de procesos. Estas restricciones se desarrollan en base a la "importancia" del proceso que se debe ejecutar en unos instantes predeterminados y predecibles, de forma que se sepa de manera fiable que dichos procesos realizan su función dentro de los límites temporales establecidos. Para realizar esta función, lo primero es conocer que es lo que se entiende por sistema en tiempo real y las diferentes categorías en las que se pueden dividir, de forma que se disponga de una separación clara entre procesos prioritarios y no prioritarios, en cuanto a ejecución temporal se refieren.

Como consecuencia de la prioridad de las tareas, es muy importante definir el concepto de fiabilidad, que nos permita asegurar que las tareas se ejecutan en tiempo y forma, haciendo este sistema en tiempo real seguro y solvente a fallos. La tolerancia a fallos permite describir la forma de manejar posibles fallos y las posibles acciones que se deben realizar bajo distintos tipos de fallos, con el objetivo de que el sistema en tiempo real puede seguir realizando su función.

Sin embargo, no solo se debe asegurar que las tareas se ejecuten de una manera específica y predecible, sino que tampoco existan interferencias entre los procesos de comunicación de información entre dichas tareas. Esto es, las tareas deben intercambiar información entre ellas y por tanto se crea un escenario de concurrencia que debe ser resuelto con los mecanismos conocidos (variables compartidas, semáforos, mensajes, monitores, etc.) y teniendo en cuenta las propias restricciones de la planificación de sistemas de tiempo real.

Una vez que se tienen en cuenta los aspectos anteriores se pueden usar diversos tipos de estrategias de planificación, de acuerdo a los requerimientos del sistema en tiempo real que se desea desarrollar. Todos estos tipos de planificación requieren de la definición de un mecanismo de temporización, denominado reloj, que se usa como base para el desarrollo de los distintos tipos de planificación y que el sistema operativo que ejecuta el software debe proporcionar.

Teniendo en cuenta lo anterior, se pretende que esta asignatura sirva como curso introductorio a la programación de sistemas en tiempo real y tiene como objetivo principal que el alumno pueda adquirir los conocimientos y habilidades necesarias para poder desarrollar ejemplos sencillos de planificaciones temporales basadas en restricciones de tiempo real y que sea capaz de disponer de información suficiente para analizar las diferentes alternativas de programación de sistemas de tiempo real.



3.REQUISITOS PREVIOS REQUERIDOS PARA CURSAR LA ASIGNATURA

Se considera imprescindible para la realización y seguimiento del curso, que el alumno posea unos sólidos fundamentos en el área de la programación con Java. Durante la práctica totalidad del temario de la asignatura, los ejemplos prácticos y actividades planificadas presuponen un conocimiento medio del lenguaje de programación Java, es decir, la capacidad de desarrollar clases Java que implementen diferentes comportamientos y usen diferentes librerías del propio lenguaje. En particular, la asignatura enseña a emplear ciertas librerías específicas adecuadas para entornos de sistemas de tiempo real. Se presupone que el alumno ha cursado la asignatura FUNDAMENTOS DE INFORMÁTICA, ubicada en cursos anteriores del plan de estudio.

Es recomendable, aunque no necesario, el conocimiento sobre sistemas operativos. Los propios sistemas operativos están pensados como piezas de software que pueden estar distribuidas en diferentes recursos físicos y que se encargan de la ejecución de las tareas/procesos involucrados en los sistemas de tiempo real. De hecho, la propia programación de los sistemas operativos en tiempo real es un buen ejemplo de desarrollo de servicios orientados a tiempo real.

4.RESULTADOS DE APRENDIZAJE

Teniendo en cuenta la presentación y contextualización previas, los objetivos básicos de la asignatura, expresados como resultados básicos del aprendizaje, son:

- Programar sistemas de tiempo real
- Ser capaz de detectar y evaluar las situaciones prácticas derivadas del uso de la concurrencia de tareas.
- Reconocer las situaciones específicas que necesitan servicios redundantes en la implementación de aplicaciones informáticas de tiempo real y aplicar las técnicas y algoritmos adecuados.
- Analizar los efectos específicos del uso de información compartida en un entorno concurrente y explicar los posibles errores de programación que pueden aparecer en los sistemas de tiempo real.

Los resultados de aprendizaje están relacionados con las competencias cognitivas específicas (CE.04, CE.07) de la titulación de grado.

5.CONTENIDOS DE LA ASIGNATURA

La asignatura está dividida en tres módulos que abarcan los diferentes aspectos relevantes de la programación de sistemas de tiempo real: definición de un entorno de tiempo real y tolerancia a fallos, los aspectos derivados de la propia concurrencia de tareas en sistemas de tiempo real y la planificación y gestión de la temporización de ejecución del software asociado a un sistema de tiempo real. El índice detallado para cada uno de los módulos se detalla a continuación:

MÓDULO 1: CONCEPTOS BÁSICOS DE LOS SISTEMAS EN TIEMPO REAL

Este módulo explica los fundamentos de los sistemas de tiempo real y las características a tener en cuenta en su desarrollo. Se define el modelo de fallo y el concepto de tolerancia a fallos, indicando algunas técnicas para su resolución. Se termina el módulo explicando como se gestionan las excepciones (posibles errores) desde el punto de vista del lenguaje de programación y del propio sistema operativo (en tiempo real o no)

Unidad 1. Introducción a los sistemas en tiempo real

- 1.1. Definición de un sistema de tiempo real
- 1.2. Ejemplos de sistemas en tiempo real
- 1.3. Características de los sistemas de tiempo real
- 1.4. Ciclo de desarrollo para sistemas en tiempo real



- 1.5. Lenguajes para programar sistemas en tiempo real

Unidad 2. Fiabilidad y tolerancia a fallos

- 2.1. Fiabilidad y fallos
- 2.2. Modos de fallo
- 2.3. Prevención de fallos y tolerancia
- 2.4. Programación N-version
- 2.5. Redundancia dinámica del software
- 2.6. La aproximación de recuperación de bloques para la tolerancia a fallos software
- 2.7. Comparación entre N-version y recuperación de bloques
- 2.8. Redundancia dinámica y excepciones
- 2.9. Medición y Predicción de la fiabilidad del software
- 2.10. Seguridad, fiabilidad y dependencia

Unidad 3. Excepciones y manejo de excepciones

- 3.1. Gestión de las excepciones en lenguajes de tiempo real antiguos
- 3.2. Gestión de excepciones en la actualidad
- 3.3. Excepciones en Ada, Java y C
- 3.4. Recuperación de bloques y excepciones

MÓDULO 2: ASPECTOS DE LA CONCURRENCIA EN SISTEMAS EN TIEMPO REAL

Este módulo explica los conceptos de ejecución concurrente, que se basa en el modelo de procesos/tareas/hebras. Se indica cómo se implementa la concurrencia en los tres modelos de programación (Java RT, Ada y C/Real Time POFIX) para pasar a describir como se implementan los mecanismos de comunicación y sincronización mediante variables compartidas y mensajes. A continuación se introduce el concepto de acción atómica y la forma de implementarlas en los tres modelos de programación propuestos. Se finaliza el módulo explicando la gestión de acceso a los recursos y el modelo de interbloqueo.

Unidad 4. Programación concurrente

- 4.1. Procesos y tareas/hebras
- 4.2. Ejecución concurrente
- 4.3. Representación de una tarea
- 4.4. Ejecución concurrente en Ada
- 4.5. Ejecución concurrente en Java
- 4.6. Ejecución concurrente en C/Real-Time POSIX
- 4.7. Sistemas con múltiples procesadores y distribuidos



4.8. Un sistema embebido simple

4.9. Concurrencia soportada por el lenguaje y soportada por el sistema operativo

Unidad 5. Sincronización y comunicación basada en variables compartidas

5.1. Exclusión mutua y condición de sincronización

5.2. Espera de recurso ocupado

5.3. Suspender y activar

5.4. Semáforos

5.5. Regiones críticas condicionales

5.6. Monitores

5.7. Variables de condición en C/Real-Time POSIX

5.8. Objetos protegidos en Ada

5.9. Métodos sincronizados en Java

5.10. Memoria compartida en multiprocesadores

5.11 Revisión del sistema simple embebido

Unidad 6. Sincronización y comunicación basada en mensajes

6.1. Sincronización de procesos

6.2. Nombramiento de tareas y estructura de mensajes

6.3. Paso de mensajes en Ada

6.4. Espera selectiva

6.5. La sentencia select de Ada

6.6. Espera selectiva, no determinista y primitivas de sincronización

6.7. Colas de mensajes en C/Real-Time POSIX

6.8. Sistemas distribuidos

6.9. Revisión del sistema simple embebido

Unidad 7. Acciones atómicas, procesos concurrentes y fiabilidad

7.1. Acciones atómicas

7.2. Acciones atómicas en C/Real-Time POSIX, Ada y Real-Time Java

7.3. Acciones atómicas recuperables

7.4. Notificación asíncrona

7.5. Notificación asíncrona en C/Real-Time POSIX



7.6. Notificación asíncrona en Ada

7.7. Notificación asíncrona en Real-Time Java

Unidad 8. Control de acceso a los recursos

8.1. Control de recursos y acciones atómicas

8.2. Gestión de recursos

8.3. El poder expresivo y la facilidad de uso

8.4. El servicio de re-encolamiento

8.5. Nombramiento asimétrico y seguridad

8.6. Uso de los recursos

8.7. Interbloqueo

MODULO 3: TEMPORIZACIÓN DE LAS TAREAS DE UN SISTEMA EN TIEMPO REAL

Este módulo explica los mecanismos de temporización que se pueden aplicar en un sistema de tiempo real, tomando como base el reloj del sistema en tiempo real. Una vez definidos los conceptos necesarios (timeouts, ámbitos temporales, etc.) se explica la forma de categorizar las tareas para su inclusión y priorización en la programación temporal del sistema. Una vez hecho esto, se pueden aplicar varias estrategias de planificación (PFS, FPS, etc.) basadas en prioridades o tiempos de ejecución. Se termina el módulo mostrando las posibles soluciones a los problemas de fallos de temporización en la ejecución de las tareas (rebasamiento) y cuáles pueden ser las mejores estrategias para tener en cuenta estos posibles fallos.

Unidad 9. Servicios de tiempo real

9.1. La noción del tiempo

9.2. Acceso al reloj

9.3. Retardando una tarea

9.4. Programación de timeouts

9.5. Especificaciones temporales

9.6. Ámbitos temporales

9.7. Tareas de tiempo real

9.8. Programación de tareas periódicas

9.9. Programación de tareas aperiódicas y esporádicas

9.10. El rol de los eventos de tiempo real y sus manejadores

9.11. Control del jitter de entrada y salida

9.12. Otras aproximaciones para soportar los ámbitos temporales

Unidad 10. Planificación de sistemas de tiempo real



- 10.1. Aproximación de ciclo ejecutivo
- 10.2. Planificación basada en tareas
- 10.3. Planificación fijada por prioridad (FPS)
- 10.4. Pruebas de planificabilidad basada en la utilización para FPS
- 10.5. Análisis del tiempo de respuesta para FPS
- 10.6. Tareas esporádicas y aperiódicas
- 10.7. Sistemas de tareas con $D < T$
- 10.8. Interacciones entre tareas y bloqueo
- 10.9. Protocolos de techo de prioridad
- 10.10. Un modelo extensible de tareas para FPS
- 10.11. Planificación de primera finalización de la siguiente tarea (EDF)
- 10.12. Sistemas dinámicos y análisis al vuelo
- 10.13. Tiempo de ejecución del peor caso
- 10.14. Planificación en multiprocesadores
- 10.15. Planificación de sistemas power-aware
- 10.16. Incorporación de los overheads del sistema

Unidad 11. Programación de bajo nivel

- 11.1. Mecanismos hardware de entrada/salida
- 11.2. Requisitos del lenguaje
- 11.3. Ada
- 11.4. Java para tiempo real
- 11.5. C y otros lenguajes de tiempo real primitivos
- 11.6. Planificación de controladores de dispositivos
- 11.7. Gestión de memoria

6.EQUIPO DOCENTE

- [RAFAEL PASTOR VARGAS](#)
- [ROBERTO HERNANDEZ BERLINCHES](#)
- [PABLO RUIPEREZ GARCIA](#)

7.METODOLOGÍA Y ACTIVIDADES DE APRENDIZAJE



De forma resumida la metodología docente se concreta en:

- Adaptada a las directrices del EEES.
- La asignatura no tiene clases presenciales. Los contenidos teóricos se impartirán a distancia, de acuerdo con las normas y estructuras de soporte telemático de la enseñanza en la UNED. Esta asignatura se impartirá a distancia, utilizando una plataforma de educación a través de Internet. Se organizarán foros de discusión para dudas y debates.
- El material docente incluye cuestionarios de autoevaluación sobre los contenidos de cada módulo y distintos tipos de actividades relacionadas con la asignatura: consulta bibliográfica, consulta de información en Internet, trabajos de análisis y resumen, y uso avanzado de herramientas software.
- Las actividades de aprendizaje se estructuran en torno al estado del arte en cada una de las materias del curso y a los problemas en los que se va a focalizar en los trabajos de evaluación a distancia.

La metodología docente se desarrolla de acuerdo con los siguientes principios:

- Además de adoptar la metodología docente general del grado, y en coherencia con el propósito de utilizar los sistemas interactivos de educación con fines pedagógicos y/o formativos, la asignatura diseñada se apoya en gran medida en los recursos educativos de este medio.
- La metodología del trabajo de la asignatura se basa en una planificación temporal de las actividades. Existirán diferentes módulos o unidades didácticas. Cada uno de éstos tendrá asociado unas unidades de aprendizaje y un material asignado (capítulos del libro base, artículos relacionados, direcciones adicionales de Internet, o cualquier otro material que se proporcione). Se asignará un período para cada módulo, en el que deberán realizar las actividades relacionadas con el mismo.

8.EVALUACIÓN

La estructura de aprendizaje se basará en el trabajo autónomo del alumno (vía evaluación continua) y la realización de una prueba presencial. A continuación se describen los elementos principales que se deben conocer a la hora de saber cómo será evaluada la asignatura.

Evaluación continua

La evaluación continua supone un aspecto muy importante dentro de un proceso general de aprendizaje. En esta asignatura, se plantearán a los alumnos un proceso de evaluación continua basado en la realización de cuatro pruebas de evaluación a distancia. Para ello, en la plataforma de aprendizaje se añadirán las tareas correspondientes con la información necesaria para la elaboración de dichas pruebas. El peso total de la evaluación continua sobre la nota final será del 40% sobre el total de la nota asignada en la evaluación final, y se distribuirán los porcentajes individuales de cada prueba para completar dicho 40%. Las pruebas a desarrollar son las siguientes:

- a) Instalación y configuración de un sistema operativo en tiempo real. Estará ponderada con un 5%.
- b) Instalación y configuración de una máquina virtual Java específica para tiempo real. Estará ponderada con un 5%.
- c) Desarrollo de una aplicación simple en tiempo real sobre los entornos instalados/configurados en las pruebas anteriores. Estará ponderada con un 10%.



d) Integración de la aplicación en tiempo real con código de programación no de tiempo real. Estará ponderada con un 20%.

La evaluación final de la asignatura se llevará a cabo a partir de las siguientes pruebas:

- Calificación asociada a las pruebas de evaluación continua, es necesario aprobarla para la superación de la asignatura.
- Realización de un examen teórico.

El examen constará de un conjunto de preguntas tipo test (entre 20 y 30) y podrá incluir preguntas acerca de las pruebas de evaluación a distancia a desarrollar.

Criterios de Evaluación

Para que un alumno pueda aprobar la asignatura deberá haber superado el examen teórico (un cinco sobre diez, como mínimo) y haber aprobado la práctica obligatoria. La nota final se calcula de la siguiente manera

$$\text{Nota final} = 0.6 \times \text{Nota_Examen} + 0.4 \times \text{Nota_Evaluacion_Continua}$$

Solo se tendrán en cuenta aquellas calificaciones que tengan aprobadas tanto el examen como las pruebas de evaluación continua a distancia.

9.BIBLIOGRAFÍA BÁSICA

ISBN(13): 9780321417459
Título: REAL-TIME SYSTEMS AND PROGRAMMING LANGUAGES (4)
Autor/es: Andrew J Wellings ; Burns, Alan ;
Editorial: ADDISON-WESLEY

Buscarlo en Editorial UNED

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Comentarios y anexos:

El libro base de Burns recoge todo del temario de la asignatura y presenta de una forma clara y concisa todos los conceptos clave. En cada capítulo se introducen los conceptos de una manera sencilla, progresiva y acompañados de ejemplos aclaratorios finalizando con un resumen que recoge los conceptos fundamentales presentados al alumno. Adicionalmente se le entregarán al estudiante una colección de enlaces y recursos relevantes que completarán la bibliografía básica con contenidos de actualidad sobre la programación de sistemas de tiempo real

10.BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13): 9780137142989
Título: REAL-TIME JAVA PROGRAMMING: WITH JAVA RTS (Primera edición)
Autor/es: Greg Bollella ; Eric J. Bruno ;
Editorial: Ed. Prentice-Hall

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED



Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780201633924

Título: PROGRAMMING WITH POSIX THREADS (Primera edición)

Autor/es: D. Butenhof ;

Editorial: ADDISON - WESLEY

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780470844373

Título: CONCURRENT AND REAL-TIME PROGRAMMING IN JAVA (Primera edición)

Autor/es: Andrew J Wellings ;

Editorial: JOHN WILEY & SONS

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780521866972

Título: CONCURRENT AND REAL-TIME PROGRAMMING IN ADA (Tercera Edición)

Autor/es: Andrew J Wellings ; Burns, Alan ;

Editorial: CAMBRIDGE UNIVERSITY PRESS

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9781419656491

Título: REAL-TIME JAVA PLATFORM PROGRAMMING (Segunda edición)

Autor/es: Peter C Dibble ;

Editorial: BookSurge Publishing

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED



Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9788478290581

Título: SISTEMAS DE TIEMPO REAL Y LENGUAJES DE PROGRAMACIÓN (3ª)

Autor/es: Wellings, Andy ; Burns, Alan ;

Editorial: PEARSON ADDISON-WESLEY

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

Comentarios y anexos:

El estudiante puede consultar la siguiente bibliografía con el fin de aclarar o extender los conocimientos que debe adquirir a lo largo del curso, y más en concreto en lo concerniente a Java RTS, Ada y POSIX para la realización de las actividades prácticas:

- Sistemas de tiempo real y lenguajes de programación. Alan Burns y Andy Wellings. Editorial: Addison-Wesley, 3ª Edición (2003). ISBN: 84-7829-058-3

Edición anterior del libro de texto básico de la asignatura. Se puede emplear como referencia adicional y apoyo de lectura del libro revisado (edición en inglés).

- Real-Time Java Programming: With Java RTS. Eric J. Bruno y Greg Bollella. Editorial: Prentice Hall (2009). ISBN: 978-0137142989

El libro se centra en la implementación de Java RTS, como motor de desarrollo para sistemas de tiempo real, incluyendo capítulos específicos sobre el recolector de basura, eficiencia de la implementación y detalles sobre la planificación y gestión temporal de las tareas, además de los modelos de memoria para RTSJ

- Real-Time Java Platform Programming (Segunda Edición). Dr. Peter C Dibble. Editorial: BookSurge Publishing (2008). ISBN: 978-1419656491

El libro describe los aspectos relevantes de la especificación RTJS, focalizando en temas como la integración con tareas de tipo Non-RT (no real time), accesos a los diferentes tipos de heap de memoria, uso de la memoria "sucia" para acceder a distintos dispositivos hardware o detallar los errores más habituales en el uso de Java RT para el desarrollo de aplicaciones.

- Concurrent and Real-Time Programming in Java. Andrew Wellings. Editorial: Willey (2004). ISBN: 978-0470844373

El libro se centra en los aspectos de programación concurrente que proporciona el modelo RTJS, y en particular en el modelo de gestión de memoria y control de recursos compartidos. Dispone de un ejemplo



práctico muy significativo, en el área de la automoción, presentando un caso de desarrollo de un sistema de control de crucero.

- Concurrent and Real-Time Programming in Ada. Alan Burns y Andy Wellings. Editorial: Cambridge University Press (Tercera Edición) (2007). ISBN: 978-0521866972

Se presenta el lenguaje de programación ADA, y en concreto las características que proporciona para la programación de sistemas en tiempo real y computación distribuida. ADA es un lenguaje pensado para ser escalable y robusto, con una fuerte estructura de planificación adecuada para los sistemas de tiempo real. Se muestran los mecanismos proporcionados para la gestión de memoria compartida y la planificación de tiempos de las tareas.

- Programming with POSIX threads. D. Butenhof. Editorial: Addison-Wesley (1997). ISBN: 978-0201633924

En este libro se describe el modelo de tareas (threads) usados en los sistemas Unix (POSIX) y que formalmente se denomina como Pthreads (de POSIX threads). Se describen los mecanismos de definición e intercomunicación de dichas tareas y su aplicabilidad al desarrollo de sistemas de computación distribuidos y de tiempo real.

11. RECURSOS DE APOYO

Curso virtual

Para alcanzar todos los objetivos propuestos, el curso se va a articular, como ya se ha comentado, a través de una plataforma especialmente diseñada para facilitar el trabajo colaborativo en Internet (basada en comunidades virtuales), desarrollada por la Sección de Innovación del Centro de Innovación y Desarrollo Tecnológico de la UNED: aLF, ubicada en <http://www.innova.uned.es>.

La plataforma de e-Learning aLF, proporcionará el soporte requerido para gestionar los procesos de enseñanza y aprendizaje, compartir documentos y enlaces de interés, crear y participar en comunidades temáticas y grupos de trabajo específicos, realizar proyectos de diversa naturaleza, organizar el trabajo mediante agendas compartidas e individuales, acceder y publicar noticias de interés, etc.

La plataforma de aprendizaje en Internet permitirá realizar el seguimiento de las actividades del curso, así como estar al tanto de cualquier información o documentación de interés relacionada con el mismo. Para poder utilizar esta plataforma y para mantener un contacto personal con el alumnado se necesitará una dirección de correo electrónico suministrada por el Centro de Servicios Informáticos de la Uned. La filosofía de uso es bien sencilla. Todas las interacciones se hacen a través de enlaces. Por lo tanto, con sólo seguir dichos enlaces se podrá acceder a foros de discusión, documentos de compañeros, etc.

Una vez familiarizados con su uso, es importante tener en cuenta que todas las novedades, instrucciones, actividades se van a publicar utilizando este medio, por tanto, el alumno debe entrar en el grupo frecuentemente para ver si hay alguna novedad en el curso. Si, además, tiene activados ciertos avisos, podrá recibir notificaciones en el correo electrónico utilizado para acceder a la plataforma de los mensajes publicados en los foros, los documentos subidos, las citas puestas en el calendario, por lo que tendrá una información instantánea de todo lo que acontece en la plataforma.

Se ofrecerán las herramientas necesarias para que, tanto el equipo docente como el alumnado, puedan compaginar el trabajo individual y el aprendizaje colaborativo.

Software para prácticas.

Se ubicará en la propia plataforma, en el área correspondiente, o bien se darán los enlaces correspondientes de las ubicaciones originales donde descargar tanto el software como los correspondientes manuales.

12. TUTORIZACIÓN



La tutorización de los estudiantes tendrá lugar esencialmente a través de los foros de la plataforma, aunque también podrán utilizarse ocasionalmente otros medios, tales como chats interactivos, servicios de mensajería instantánea y el correo electrónico. Adicionalmente, está también previsto, para temas personales que no afecten al resto de los estudiantes, atender consultas en persona o por teléfono.

El seguimiento del aprendizaje se realizará revisando la participación de los alumnos en los distintos foros de debate y las aportaciones de material nuevo además de la entrega en fecha de los diferentes trabajos prácticos que se han planificado durante la evolución del curso.

Adicionalmente a la participación en el curso virtual, se puede contactar con el equipo docente en horario de guardia (por teléfono) o a través del correo electrónico:

Rafael Pastor Vargas, Lunes de 16:00 a 18:00 horas (rpastor@scc.uned.es)

Salvador Ros Muñoz, Jueves de 15:00 a 17:00 horas (sros@scc.uned.es)

Roberto Hernández Berlinches, Lunes de 16:00 a 18:00 horas (roberto@scc.uned.es)

Para el contacto por correo electrónico se debe comenzar el Asunto del mensaje mediante el prefijo [STR], es decir, un mensaje de ejemplo como consulta al equipo docente tendría como asunto del correo:

[STR] Duda sobre el temario

