



FICHA TÉCNICA DE LA ASIGNATURA

Datos de la asignatura	
Nombre completo	Fundamentos de los Sistemas operativos
Código	DTC-IMAT-215
Título	Grado en Ingeniería Matemática e Inteligencia Artificial
Impartido en	Grado en Ingeniería Matemática e Inteligencia Artificial [Segundo Curso]
Nivel	Reglada Grado Europeo
Cuatrimestre	Semestral
Créditos	6,0 ECTS
Carácter	Obligatoria (Grado)
Departamento / Área	Departamento de Telemática y Computación
Responsable	Pablo Sánchez Pérez
Horario	Mañana
Horario de tutorías	A concertar directamente con el profesor.

Datos del profesorado	
Profesor	
Nombre	Antonio García de Burgos Velón
Departamento / Área	Departamento de Telemática y Computación
Correo electrónico	agarciaburgos@icai.comillas.edu
Profesor	
Nombre	Pablo Sánchez Pérez
Departamento / Área	Departamento de Telemática y Computación
Despacho	419
Correo electrónico	psperez@icai.comillas.edu
Profesores de laboratorio	
Profesor	
Nombre	Israel Alonso Martínez
Departamento / Área	Departamento de Telemática y Computación
Despacho	407
Correo electrónico	ialonso@icai.comillas.edu

DATOS ESPECÍFICOS DE LA ASIGNATURA

Contextualización de la asignatura
Aportación al perfil profesional de la titulación



El objetivo principal de la asignatura es proporcionar a los alumnos los conocimientos básicos acerca de los sistemas operativos, prestando especial énfasis al sistema operativo GNU/Linux. Se estudiará los componentes hardware básicos del ordenador, incluyendo el procesador, memoria caché, memoria principal y disco, y se estudiará en más detalle la gestión de procesos, hilos, memoria y ficheros.

Al finalizar el curso, los alumnos habrán adquirido un conocimiento sólido acerca de los elementos más importantes de un sistema operativo. Además, los alumnos se habrán familiarizado con el Sistema Operativo de GNU/Linux, y habrán aprendido a ejecutar comandos y scripts para automatizar tareas. Además, serán capaces de crear aplicaciones multiprocesador y multihilo optimizando la carga computacional, aprovechando de manera eficiente el hardware del ordenador.

Prerrequisitos

Programación en Python.

Manejo de un ordenador a nivel básico.

Competencias - Objetivos

Competencias

GENERALES

CG04	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
CG05	Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería

ESPECÍFICAS

CE17	Capacidad para analizar y distinguir las características, funcionalidades y estructura de los sistemas operativos y diseñar aplicaciones basadas en sus servicios.
CE38	Capacidad para entender el funcionamiento de los ordenadores y el papel que juegan los sistemas operativos en el proceso de ejecución de programas.

Resultados de Aprendizaje

RA1	Conocer la arquitectura básica de un ordenador
RA2	Conocer las funciones y estructura de un sistema operativo
RA3	Entender el funcionamiento de un sistema operativo desde el punto de vista de usuario
RA4	Comprender el significado de un proceso y un hilo, así como las ventajas e inconvenientes del modelo de proceso monohilo o multihilo
RA5	Conocer la forma en la que los procesos son administrados y controlados por un sistema operativo
RA6	Conocer los algoritmos básicos de planificación de procesos
RA7	Conocer los diferentes esquemas de asignación de memoria



RA8	Conocer el funcionamiento de los sistemas de gestión de la memoria virtual
RA9	Conocer la gestión del almacenamiento de ficheros y directorios
RA10	Conocer los sistemas de archivos más relevantes en la actualidad
RA11	Ser capaz de diseñar e implementar aplicaciones que usen las llamadas al sistema operativo

BLOQUES TEMÁTICOS Y CONTENIDOS

Contenidos – Bloques Temáticos

El temario del curso está dividido en 6 grandes temas: Introducción general a los sistemas informáticos, Introducción al sistema operativo Linux, Red, y entornos, Gestión de procesos e hilos, Máquinas virtuales, Gestión de memoria y Gestión de entrada/salida y sistemas de archivos.

Bloque 1. Introducción general

Introducción general

Elementos básicos de un ordenador.

Funciones y estructura general de un sistema operativo.

Bloque 2. Introducción al sistema operativo Linux, Red, y entornos.

Introducción al sistema operativo Linux, Red, y entornos.

Sistema Operativo GNU/Linux.

Protocolos de red.

Gestión de Dependencias Python.

Bloque 3. Gestión de procesos

Gestión de procesos

Procesos e hilos. Características e implementación en Python.

Planificación de procesos.

Concurrencia, exclusión mutua, semáforos.

Bloque 4. Máquinas virtuales

Máquinas virtuales

Virtualización y máquinas virtuales



Introducción básica a Docker

Bloque 5. Gestión de memoria

Gestión de memoria

Requisitos y particionamiento de memoria.

Esquemas de asignación de memoria.

Memoria virtual.

Bloque 6. Gestión de entrada/salida y sistemas de archivos

Gestión de entrada/salida y sistemas de archivos

Principios, técnicas de entrada/salida.

Sistemas de archivos: archivos y directorios.

Implementación del sistema de archivos.

METODOLOGÍA DOCENTE

Aspectos metodológicos generales de la asignatura

Con el fin de conseguir el desarrollo de competencias propuesto, la materia se desarrollará teniendo en cuenta la actividad del alumno como factor prioritario. Tanto las sesiones presenciales como las no presenciales promoverán la implicación activa de los alumnos en las actividades de aprendizaje. Por lo tanto, las metodologías en las que se hará mas énfasis serán las siguientes:

- Lección magistral
- Aprendizaje práctico
- Aprendizaje basado en proyectos

Aunque también se explorarán las siguientes metodologías:

- Clase invertida
- Aprendizaje colaborativo

Metodología Presencial: Actividades

1. Lección expositiva (magistral): el profesor desarrolla el tema que previamente los alumnos deben de haber leído, explicándolo en la pizarra y en el ordenador mediante diapositivas. Se realizarán ejercicios intermedios en las clases y se fomentará la participación de los alumnos tanto en las clases como en los ejercicios propuestos.

CG04, CG05, CE38, CE17

2. Clases live-coding y aprendizaje colaborativo: el profesor indicará y enseñará a los alumnos programas para poner en práctica lo aprendido en las clases de teoría. Mientras el profesor enseña, los alumnos replicarán lo que ha ido haciendo el profesor en su ordenador personal. El profesor proporcionará a los alumnos manuales sobre distintos contenidos que los alumnos deben de haber leído antes de las clases de live-coding. Las herramientas/programas que el profesor emplee en estas clases deben haber sido previamente instaladas por los alumnos en horas no lectivas (los alumnos dispondrán de manuales de

CG04, CG05, CE38, CE17



instalación de los sucesivos programas). En algunas clases, se podrá pedir a los alumnos realizar actividades en grupos que deberán entregar posteriormente al profesor.

3. Prácticas de laboratorio y aprendizaje basado en el proyecto: a lo largo del curso los alumnos realizarán prácticas individuales de cada uno de los temas expuestos. Empezarán a trabajar en casa sobre la práctica y en el laboratorio resolverán las dudas que puedan tener finalizando la práctica. Por otro lado, los alumnos deberán realizar un proyecto que consistirá en aplicar los conceptos vistos a lo largo de toda la asignatura. Para el proyecto, se dedicarán algunas horas presenciales, pero deberá realizarse mayoritariamente en horas no presenciales.

CG04, CG05, CE38, CE17

Metodología No presencial: Actividades

1. Estudio teórico:

- a. Estudio y preparación de los temas o conceptos que vayan a ser expuestos por el profesor.
- b. Estudio individual y personal por parte del alumno de los conceptos ya expuestos en las lecciones expositivas.

El objetivo principal del estudio teórico no presencial es alcanzar una comprensión más profunda de los conceptos teóricos de la asignatura que han sido explicados previamente por los profesores durante las clases.

CG04, CG05, CE38, CE17

2. Casos prácticos:

Preparación y comienzo del desarrollo de las prácticas de laboratorio propuestas semanalmente por el profesor.

El objetivo principal del trabajo no presencial de los casos prácticos es ser capaz de dar respuesta a las prácticas propuestas por los profesores incidiendo no sólo en la funcionalidad sino también en el estilo de código (cuando sea necesario).

CG04, CG05, CE38, CE17

3. Aprendizaje basado en proyectos:

Los alumnos además realizarán en las horas no lectivas un proyecto aplicando los conocimientos adquiridos a lo largo del curso.

El objetivo del proyecto es que los alumnos puedan aplicar los conocimientos adquiridos en la asignatura en una aplicación más compleja teniendo en cuenta un mayor número de requerimientos.

CG04, CG05, CE38, CE17

RESUMEN HORAS DE TRABAJO DEL ALUMNO

HORAS PRESENCIALES				
Clases magistrales expositivas y participativas	Sesiones prácticas con uso de software	Tutorías para resolución de dudas	Ejercicios prácticos y resolución de problemas	Actividades de evaluación continua del rendimiento



33.00	21.00	7.00	3.00	3.00
HORAS NO PRESENCIALES				
Sesiones prácticas con uso de software	Estudio personal			
49.00	50.00			
CRÉDITOS ECTS: 6,0 (166,00 horas)				

EVALUACIÓN Y CRITERIOS DE CALIFICACIÓN

Actividades de evaluación	Criterios de evaluación	Peso
<ul style="list-style-type: none">Intersemestral: 20%Final: 50%	<p>Prueba Intersemestral (20%): examen orientado a comprender de los conceptos teóricos de los fundamentos de los sistemas operativos (incluyendo la arquitectura básica de un ordenador), comandos Linux, scripting de bash, gestión de procesos y planificación de procesos (si da tiempo).</p> <p>Prueba Final (50%): examen conteniendo el temario de la prueba intersemestral ampliándolo con conceptos adicionales relacionados con la gestión de procesos, virtualización y gestión de memoria y ficheros.</p>	70 %
<ul style="list-style-type: none">Proyecto final: 15%	<p>Proyecto final (15%): Se evaluará además de la funcionalidad, el diseño empleado en el proyecto, la limpieza y el estilo de código, y la aplicación del temario de la asignatura.</p>	15 %
<ul style="list-style-type: none">Prácticas semanales: 15%	<p>Prácticas semanales (15%): Las prácticas estarán orientadas a profundizar en los conceptos explorados en las clases de teoría resolviendo ejercicios planteados en un enunciado de forma individual. Se evaluará además de la funcionalidad, el estilo de código empleado en las prácticas y la calidad de los informes solicitados.</p>	15 %

Calificaciones

Hay dos convocatorias. Convocatoria ordinaria en diciembre/enero y extraordinaria en junio.

Calificación Convocatoria Ordinaria



La calificación de la **Convocatoria Ordinaria** (CO) será la siguiente:

- Un 50% el examen final de la asignatura (EX_F)
- Un 20% la prueba intersemestral (EX_I)
- Un 15% el proyecto (PROY)
- Un 15% las prácticas semanales (PRACT)

Es decir, la nota de la convocatoria ordinaria será la siguiente:

$$CO = 0.5 * EX_F + 0.2 EX_I + 0.15 * PROY + 0.15 * PRACT$$

Será necesario que $CO \geq 5$ para aprobar la asignatura y se deben cumplir las siguientes condiciones:

- Será obligatorio que $EX_F \geq 5$. Es decir, la nota del examen final de la signatura debe ser igual o superior a 5. En caso contrario, $CO = EX_F$.
- Será obligatorio que $(PROY + PRACT) / 2 \geq 5$. Es decir, se debe obtener una nota mayor o igual a 5 en la media obtenida entre el proyecto y las prácticas semanales tanto en convocatoria extraordinaria como en ordinaria para aprobar la asignatura. En caso de no llegar a esa nota mínima en convocatoria ordinaria, podrán entregarse de nuevo tanto el proyecto como las prácticas en convocatoria extraordinaria. En el supuesto que esta condición no se consiga, $CO = (PROY + PRACT) / 2$. La nota del examen en la convocatoria ordinaria se guarda para la convocatoria extraordinaria en caso de estar aprobado.
- Será necesario obtener una nota de al menos un 4 tanto en el proyecto como en la media de prácticas para aprobar la asignatura. Es decir, $PROY \geq 4$ y $PRACT \geq 4$. En caso de no cumplir esta restricción, $CO = \text{MIN}(PRACT, PROY)$. La nota del examen en la convocatoria ordinaria se guarda para la convocatoria extraordinaria en caso de estar aprobado.

NORMAS GENERALES:

- La inasistencia al 15% o más de las horas presenciales de esta asignatura puede tener como consecuencia la imposibilidad de presentarse a las convocatorias ordinaria y extraordinaria.
- Las prácticas/proyecto entregadas con un límite de 24h después de su fecha límite tendrán una penalización del 50% en la nota. Pasadas 24h de la fecha límite, la calificación de la práctica/proyecto será de 0.

Calificación Convocatoria Extraordinaria

La calificación de la **convocatoria extraordinaria** (CE) será la siguiente:

- Un 70% el examen final de la asignatura (EX_F, no se tiene en cuenta el examen intersemestral)
- Un 15% el proyecto (PROY)
- Un 15% las prácticas semanales (PRACT)

Es decir:

$$CE = 0.7 * EX_F + 0.15 * PROY + 0.15 * PRACT$$

Será necesario que $CE \geq 5$ para aprobar la asignatura y se deben cumplir las siguientes restricciones:

- Será obligatorio que $EX_F \geq 5$. Es decir, la nota del examen final de la asignatura de la convocatoria extraordinaria debe ser igual o superior a 5. En caso contrario, $CE = EX_F$.
- Será obligatorio que $(PROY + PRACT) / 2 \geq 5$. Es decir, se debe obtener una nota mayor o igual a 5 en la media obtenida entre el proyecto y las prácticas semanales. En caso de no llegar a esa nota mínima, $CE = (PROY + PRACT) / 2$.



- Será necesario obtener una nota de al menos un 4 tanto en el proyecto como en la media de prácticas para aprobar la asignatura. Es decir, $PROY \geq 4$ y $PRACT \geq 4$. En caso de no cumplir esta condición, $CE = \min(PRACT, PROY)$.

La inasistencia al 15% o más de las horas presenciales de esta asignatura puede tener como consecuencia la imposibilidad de presentarse a las convocatorias ordinaria y extraordinaria.

NORMAS GENERALES:

- La inasistencia al 15% o más de las horas presenciales de esta asignatura puede tener como consecuencia la imposibilidad de presentarse a las convocatorias ordinaria y extraordinaria.
- Las prácticas/proyecto entregadas con un límite de 24h después de su fecha límite tendrán una penalización del 50% en la nota. Pasadas 24h de la fecha límite, la calificación de la práctica/proyecto será de 0.

PLAN DE TRABAJO Y CRONOGRAMA

Actividades	Fecha de realización	Fecha de entrega
Lectura y estudio de los contenidos teóricos, apuntes y código facilitado proporcionado por el profesor.	Después y antes de cada clase	
Proyecto final	Al menos 1 mes antes de terminar el curso	Última semana de clases
Realización de los exámenes	Octubre, Diciembre/Enero, Junio	
Realización de las prácticas semanales	Antes, durante y después de la clase de prácticas	1 semana después de su publicación en moodle
Bloque 1. Introducción general	Al inicio del curso	1,5 semanas aproximadamente
Bloque 2. Introducción al sistema operativo Linux, Red, y entornos.	Después del Tema 1. Durante el mes de septiembre	2 semanas y media aproximadamente
Bloque 3. Gestión de procesos	Después del Tema 2. Finales de septiembre. Mes de octubre	4 semanas aproximadamente
Bloque 4. Máquinas virtuales	Después de terminar el tema 3. Mes de noviembre	1 semana aproximadamente
Bloque 5. Gestión de memoria.	Después del tema 4. Segunda mitad de noviembre	2,5 semanas aproximadamente
Bloque 6. Gestión de entrada/salida y sistemas de archivos.	Al terminar el tema 5. Mes de diciembre.	1 semana aproximadamente

BIBLIOGRAFÍA Y RECURSOS

Bibliografía Básica

Moodle (manuales y transparencias del profesor)



Bibliografía Complementaria

Aspectos internos y principios de diseño, de W. Stallings Ed: Pearson. (Novena edición)

Sistemas Operativos Modernos, de A. Tanenbaum. Ed: Pearson Educacion

Fundamentos de Sistemas Operativos. A. Silberschatz, P. Baer Galvin, G. Gagne. Ed: McGraw Hill

Además de los manuales que se podrán consultar en moodle, se recomienda complementar dicha documentación con información disponible en Internet:

- <https://www.tutorialspoint.com/unix/unix-useful-commands.htm> → Para comandos linux.
- <https://www.tutorialspoint.com/awk/index.htm> → Para awk.
- <https://superfastpython.com/threading-in-python/> → Para crear Porcesos e Hilos en Python.
- <https://docs.python.org/es/3/howto/sockets.html> → Para programación de sockets en Python.

En cumplimiento de la normativa vigente en materia de **protección de datos de carácter personal**, le informamos y recordamos que puede consultar los aspectos relativos a privacidad y protección de datos [que ha aceptado en su matrícula](#) entrando en esta web y pulsando "descargar"

<https://servicios.upcomillas.es/sedelectronica/inicio.aspx?csv=02E4557CAA66F4A81663AD10CED66792>