
Curso Académico: (2024 / 2025)**Fecha de revisión: 23-04-2024**

Departamento asignado a la asignatura: Departamento de Ingeniería Telemática**Coordinador/a: MORENO MARCOS, PEDRO MANUEL****Tipo: Obligatoria Créditos ECTS : 6.0****Curso : 2 Cuatrimestre : 1**

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

Se recomienda haber superado tanto Programación como Programación de Sistemas, y Electrónica Digital

OBJETIVOS

1. Ser capaz de diseñar un sistema software. El sistema debe contener estructuras de datos no triviales, gestión dinámica de memoria, procesos y comunicación entre procesos y utilizar técnicas de ingeniería para traducir un conjunto de restricciones de alto nivel, derivadas de un hipotético escenario industrial, en una aplicación robusta.
2. Ser capaz de comprender el concepto de proceso e hilo, entender distintos problemas de concurrencia y aplicar para cada caso los mecanismos de sincronización adecuados.
3. Ser capaz de utilizar con soltura las siguientes herramientas utilizadas en entornos industriales: un compilador con las opciones para generar diferentes versiones para depuración y analizar los mensajes que se obtienen mientras se desarrolla una aplicación, un programa de control de versiones para manejar los flujos de desarrollo y herramientas de análisis del comportamiento de memoria en una aplicación.
4. Ser capaz de: trabajar de forma efectiva tanto de forma individual como en equipo, aprender de forma autónoma, aprender de forma autónoma, manipular las diferentes fuentes de información, generar información concisa sobre las tareas conseguidas, manejar el tiempo de trabajo personal, y presentar de forma efectiva los resultados derivados de su trabajo.

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

1. El lenguaje de programación C
 - Estructuras de datos básicas y construcciones de flujo
 - Estructura de una aplicación C. El pre-procesador, división de ficheros, creación de un ejecutable
 - Funciones (con varios parámetros, con parámetros variables)
 - Manipulación de punteros y punteros a funciones
 - Funciones de entrada/salida con ficheros
2. Gestión de memoria dinámica en C
 - Estructuras de datos dinámicas
 - Fugas de memoria
 - Herramientas para detectar fugas de memoria
3. Sistemas Operativos
 - Procesos e hilos
 - Creación de procesos
 - Comunicación entre procesos: tuberías y señales
 - Concurrencia y mecanismos de sincronización: cerrojos, semáforos y monitores

- 4. Diseño de una aplicación
 - Análisis de requisitos
 - Descomposición de funcionalidad a alto nivel
 - Diseño descendente

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Las actividades utilizadas para verificar las competencias y destrezas en el curso son:

- Ejercicios sobre los siguientes temas: diseño de las estructuras de datos más apropiadas para una funcionalidad en una aplicación, escribir fragmentos de código para manipular estructuras de datos, lectura/escritura de campos de las estructuras, datos sobre los procesos, creación y comunicación de procesos e hilos etc; cálculo de la cantidad de memoria ocupada por diferentes estructuras de datos.
- Durante las sesiones de laboratorio se escriben, compilan, enlazan y ejecutan fragmentos de código con las diferentes opciones del compilador para incluir información de depuración, definir constantes, etc. Se analiza la corrección de estos fragmentos de código con el depurador.
- Durante las sesiones de laboratorio se escriben fragmentos de código para crear, destruir y manipular estructuras de datos utilizando memoria dinámica. También se dividirá la funcionalidad del programa requerido entre múltiples funciones de las que escriben su código.
- Durante las sesiones de laboratorio, se analizarán los requisitos derivados de las especificaciones proporcionadas en ejercicios y se trabajará en el diseño de la solución.
- Búsqueda de documentos auxiliares para completar la información que se estudia en un tema.
- Uso de las siguientes herramientas: máquinas virtuales, compilador y controlador de versiones.

Durante estas actividades la plantilla docente revisa el trabajo de los estudiantes en la clase, supervisa las sesiones de laboratorio, responde a las preguntas de los estudiantes, mantiene una hora de tutoría semanal y convoca tutorías con múltiples alumnos cuando lo considera oportuno.

SISTEMA DE EVALUACIÓN

Peso porcentual del Examen Final:	50
Peso porcentual del resto de la evaluación:	50

Dado el alto porcentaje de objetivos que dependen directamente de los objetivos de las asignaturas "Programación" y "Programación de sistemas" del primer año, a no ser que se hayan aprobado ambas, NO se recomienda cursar Arquitectura de Sistemas.

La evaluación de la asignatura se desarrolla utilizando los siguientes criterios:

** Evaluación continua (50%): Consta de pruebas de evaluación parciales, en el que el alumno debe demostrar individualmente el conocimiento adquirido (teoría y problemas) para un bloque de unidades didácticas. Entre este conocimiento, se deberá demostrar haber realizado y comprendido las actividades de programación realizadas en las clases prácticas.

Primer examen parcial (25%)

Segundo examen parcial (25%)

** Examen final (50%). Sirve para demostrar el conocimiento adquirido por el estudiante durante todo el curso

Es necesario obtener una nota mínima de 3.5 en el examen final para poder superar la asignatura
La participación del alumno también podrá considerarse en la nota final de la asignatura.

Respecto al uso de herramientas de inteligencia artificial, se podrán usar en el proceso formativo, pero no en las pruebas de evaluación.

BIBLIOGRAFÍA BÁSICA

- Abraham Silberschatz Operating system concepts, John Wiley & Sons, 2002
- C. Michael Pilato, Ben Collins-Sussman y Brian W. Fitzpatrick Version Control with Subversion, 2nd Edition, O'Reilly Media, Inc., 2008
- David Griffiths y Dawn Griffiths Head First C, O'Reilly Media, Inc., 2012
- Michael Kerrisk The Linux Programming Interface, No Starch Press, 2010
- Richard Reese Understanding and Using C Pointers, O'Reilly Media, Inc., 2013
- Stephen G. Kochan Programming in C, Fourth Edition, Addison-Wesley Professional, 2014

BIBLIOGRAFÍA COMPLEMENTARIA

- Brian W. Kernighan, Dennis M. Ritchie The C programming language, Prentice Hall, 1988
- Daniel Leuck, Patrick Niemeyer Learning Java, O'Reilly Media, Inc., 2013
- Félix García Carballeira et al. El lenguaje de programación C : diseño e implementación de programas, Pearson, 2001
- Jesús Carretero Perez, Felix García Carballeira et al. Problemas resueltos de programación en lenguaje C, Paraninfo, 2004
- Joost Visser; Sylvan Rigal, Gijs Wijnholds y Zeeger Lubsen Building Software Teams, O'Reilly Media, Inc., 2016
- Steve McConnell Code Complete, Second Edition, Microsoft Press, 2004

RECURSOS ELECTRÓNICOS BÁSICOS

- Profesores . Aula Global: <https://aulaglobal.uc3m.es>
- Profesores . Aula Virtual: <https://aulavirtual.uc3m.es>