

**Nombre de la asignatura:** Sistemas Operativos II

**Créditos:** 3-2-5

### **Aportación al perfil**

Esta asignatura apoyará al alumno en la consecución de las siguientes competencias:

- Aplicar conocimientos científicos y tecnológicos en la solución de problemas en el área informática con un enfoque interdisciplinario.
- Administrar las tecnologías de la información, para estructurar proyectos estratégicos.
- Aplicar normas, marcos de referencia y estándares de calidad y seguridad vigentes en el ámbito del desarrollo y gestión de tecnologías y sistemas de información.
- Integrar las soluciones de tecnologías de información a los procesos organizacionales para fortalecer objetivos estratégicos.
- Seleccionar y utilizar de manera óptima técnicas y herramientas computacionales actuales y emergentes.
- Identificar, diseñar, desarrollar los mecanismos de almacenamiento, distribución, visualización y manipulación de la información.

### **Objetivo de aprendizaje**

- Seleccionar apropiadamente sistemas operativos que formen parte de plataformas de software aplicadas al incremento de la productividad de las organizaciones.

## **Temario**

### **1. Introducción**

- 1.1. Definición
- 1.2. Conceptos básicos
  - 1.2.1. Compartición de recursos
  - 1.2.2. Operación síncrona
  - 1.2.3. Consenso
  - 1.2.4. Exclusión mutua distribuida
  - 1.2.5. Tolerancia a fallas
  - 1.2.6. Auto estabilización
- 1.3. Tipos de sistemas operativos distribuidos
- 1.4. Estructura de red
- 1.5. Topología de red
- 1.6. Estructura de comunicación
- 1.7. Protocolos de comunicación
- 1.8. Compartición del procesador
  - 1.8.1. Multiprocesamiento simétrico
  - 1.8.2. Grupos ("clusters")
  - 1.8.3. "Grids"
- 1.9. Algoritmos de elección
  - 1.9.1. Algoritmo "Bully"
  - 1.9.2. Algoritmo "Ring"

### **2. Comunicación en los sistemas operativos distribuidos.**

- 2.1 Comunicación.
  - 2.1.1 Comunicación con cliente servidor (sockets).
  - 2.1.2 Comunicación con RPC.
  - 2.1.3 Comunicación en grupo.
  - 2.1.4 Tolerancia a fallos.
- 2.2 Sincronización.
  - 2.2.1 Relojes físicos.
  - 2.2.2 Relojes lógicos.
  - 2.2.3 Usos de la sincronización (manejo de caché, comunicación en grupo, exclusión mutua, elección, transacciones atómicas e interbloqueo).
- 2.3 Nominación.
  - 2.3.1 Características y estructuras.
  - 2.3.2 Tipos de nombres (usuario y de sistema).
  - 2.3.3 Resolución y distribución.
  - 2.3.4 Servidores y agentes de nombres.
  - 2.3.5 Mapeo de direcciones.
  - 2.3.6 Mapeo de rutas.
  - 2.3.7 Modelo de Terry.

### **3. Procesos y procesadores en sistemas distribuidos.**

- 3.1 Procesos y procesadores conceptos básicos.
- 3.2 Hilos y multihilos.

- 3.3 Modelos de procesadores.
  - 3.3.1 De estación de trabajo.
  - 3.3.2 De pila de procesadores.
  - 3.3.3 Híbrido.
- 3.4 Asignación de procesadores.
  - 3.4.1 Modelos y algoritmos con sus aspectos de diseño e implantación.
- 3.5 Coplanificación.
- 3.6 Tolerancia a fallos.
- 3.7 Sistemas distribuidos de tiempo real.

#### **4. Memoria Compartida Distribuida (MCD).**

- 4.1 Configuraciones de MCD.
  - 4.1.1 De circuitos, basados en bus, anillo o con conmutador.
- 4.2 Modelos de consistencia.
  - 4.2.1 Estricta, causal, secuencial, débil, de liberación y de entrada.
- 4.3 MCD en base a páginas.
  - 4.3.1 Diseño, replica, granularidad, consistencia, propietario y copias.
- 4.4 MCD en base a variables.
- 4.5 MCD en base a objetos.

### **Definición de las competencias específicas**

1. Identificar las funciones, evolución, organización, el proceso de arranque y las formas en que se vincula el sistema operativo con su ambiente, reconociendo estas características generales en el estudio de un sistema operativo en particular.
2. Aplicar los conceptos básicos de administración de procesos en el análisis de las características particulares de un sistema operativo.

### **Sugerencias didácticas transversales**

1. Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.
2. Relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.
3. Llevar a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, tales como: observación, análisis, la aplicación de lenguajes de programación en la solución de problemas elementales, la contrastación del comportamiento y aplicación de los conceptos entre lenguajes de programación diferentes aplicados a la solución de un mismo problema.
4. Propiciar, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
5. Propiciar el uso adecuado de conceptos y de terminología científico-tecnológica.

## **Prácticas**

1. Instalar y configurar dos o más lenguajes que soporten el paradigma de la programación concurrente, resolviendo grupalmente, catedrático y alumnos, uno o dos ejercicios de las actividades de aprendizaje.
2. Instalar y configurar dos o más lenguajes que soporten el paradigma de la programación con “scripts”, resolviendo grupalmente, catedrático y alumnos, uno o dos ejercicios de las actividades de aprendizaje.
3. Instalar y configurar uno o más lenguajes que soporten el paradigma de la programación funcional, resolviendo grupalmente, catedrático y alumnos, uno o dos ejercicios de las actividades de aprendizaje.
4. Seleccionar grupalmente, catedrático y alumnos, un lenguaje apropiado para un caso propuesto por el maestro, aplicando los criterios vistos en clase.

## **Criterios de evaluación**

La evaluación puede hacerse considerando los siguientes criterios:

1. La aplicación de exámenes de conocimientos
2. Tareas tales como ensayos, resúmenes, informes, etcétera.
3. Solución a los casos propuestos por el catedrático.
4. Ejercicios realizados en los diferentes lenguajes de programación.

### **Fuentes de información**

1. Scott, Michael L.; ***Programming Language Pragmatics***; Third edition; ed. Morgan Kaufmann Publishers; Estados Unidos; 2009.
2. Watt, David A.; Findlay, William; ***PROGRAMMING LANGUAGE DESIGN CONCEPTS***; ed. John Wiley & Sons, Ltd; Estados Unidos; 2004.
3. Loudon, Kenneth C.; ***Lenguajes de programación: principios y práctica***; segunda edición; México, 2004.
4. Watt, David A.; ***Programming Language Processors in Java: COMPILERS AND INTERPRETERS***; ed. Pearson Education Limited; Inglaterra; 2000.