

PROGRAMA DE SISTEMAS DISTRIBUIDOS

| | | | |
|----------------------------------|-------------------------------|-------------------------------|---|
| Facultad: | Ingeniería | Departamento: | Gestión de Proyectos y Sistemas |
| Código: | FPTSP26 | Asignatura: | Sistemas Distribuidos |
| Creditos: | 3 | Tipo: | <input checked="" type="checkbox"/> Obligatoria <input type="checkbox"/> Electiva |
| Carreras: | Ingeniería de Sistemas (IS) | Trimestres: | IX (IS) |
| Prerrequisito | BPTSP04 - Sistemas Operativos | Modalidad: | Presencial |
| Número de horas semanales | | | |
| En aula | Prácticas supervisadas | Laboratorio | Aprendizaje Autónomo |
| 4 | | | 4 |
| Coordinador: | Andrés Sanoja | Fecha de actualización | Nov. 2025 |

1. **Justificación:** Esta asignatura representa un elemento clave para la formación del ingeniero de sistemas, dado que los sistemas tecnológicos actuales requieren la integración y coordinación de múltiples componentes distribuidos en diferentes ubicaciones físicas. Esta área aborda los desafíos relacionados con la comunicación, sincronización, tolerancia a fallos, seguridad y escalabilidad en entornos distribuidos. El dominio de estos conceptos y técnicas es indispensable para diseñar y administrar sistemas robustos, eficientes y seguros, capaces de soportar aplicaciones críticas y de gran escala en ámbitos industriales, comerciales y científicos.

2. **Propósito:** Formar profesionales capaces de comprender y aplicar principios, modelos y algoritmos fundamentales en sistemas distribuidos, abordando problemas de comunicación, sincronización, coordinación y seguridad. Se busca que el estudiante sea competente en el diseño, implementación y evaluación de sistemas distribuidos, con una visión integrada, ética y orientada a resolver problemas relevantes en la industria y la sociedad.

3. **Objetivos**

- Analizar y explicar los modelos, arquitecturas y protocolos fundamentales en sistemas distribuidos modernos. (Objetivo 1 de la carrera)
- Aplicar técnicas de comunicación, sincronización y coordinación para resolver problemas prácticos en entornos distribuidos. (Objetivos 1 y 2 de la carrera)
- Evaluar métodos de tolerancia a fallos, recuperación y seguridad, proponiendo soluciones robustas frente a fallos y ataques. (Objetivos 2 y 4 de la carrera)
- Diseñar e implementar prototipos de sistemas distribuidos, demostrando competencias técnicas, liderazgo y trabajo en equipo. (Objetivos 3 y 4 de la carrera)

4. **Resultados de aprendizaje**

- a) **RA10 - Diseño de soluciones de ingeniería de sistemas.** Capacidad para aplicar metodologías de diseño de soluciones informáticas, integrando conocimientos técnicos y habilidades de gestión para identificar, analizar y resolver problemas o nuevos requerimientos, considerando principios de adaptabilidad a las necesidades del entorno. (N2)
- b) **RA11 - Práctica de la ingeniería.** Capacidad para aplicar metodologías de diseño de soluciones informáticas, integrando conocimientos técnicos y habilidades de gestión para identificar, analizar y resolver problemas o nuevos requerimientos, considerando principios de adaptabilidad a las necesidades del entorno. (N2)

5. **Contenido**

| Tema | Contenido | Herramientas técnicas y actividades (proyectos, trabajos, laboratorios) | Horas dedicadas |
|------|---|---|-----------------|
| 1 | Introducción de los Sistemas Distribuidos. Clasificación. Computación distribuida de alto rendimiento. Sistemas de información distribuidos. Sistemas omnipresentes. Arquitectura de los sistemas distribuidos. Estilos de arquitecturas. Middlewares. | Clase interactiva: Contextualización y conceptos básicos de los sistemas distribuidos. Se propiciará la participación de los estudiantes mediante preguntas activadoras. Estudio de Casos: Se presentarán casos reales de sistemas distribuidos (Ej. Netflix, Google, Amazon) | 10 |
| 2 | Comunicación en Sistemas Distribuidos. Comunicación entre procesos. Llamada a procedimientos Remotos (RPC). Comunicación orientada a mensajes: Mensajería transitoria simple con sockets, Mensajería transitoria avanzada, Comunicación persistente orientada a mensajes. Comunicación multicast. Multidifusión. | Clase magistral: Explicación de los diferentes modelos de comunicación. Ejercicios prácticos: Los estudiantes realizarán ejercicios de implementación de sistemas de comunicación básicos, análisis de código, y desarrollo de proyectos usando RPC. Proyecto: Los estudiantes trabajan un proyecto individual por fases donde el estudiante debe integrar la metodología Scrum. (entregas semanalmente) Herramientas: Plataformas de programación con sockets, RPC. | 10 |
| 3 | Coordinación en Sistemas Distribuidos. Transacciones y control de concurrencia. Transacciones distribuidas. ACID. Transacciones asíncronas. Gestión | Clase participativa: Presentación del concepto de coordinación, ejemplos y algoritmos fundamentales. Se fomentará la participación de los estudiantes mediante preguntas y discusiones | 10 |

| | | | |
|---|---|---|----|
| | Distribuida de Procesos. Consistencia y Replicación: Consistencia basada en datos. Consistencia basada en el cliente. Gestión de la replicación. | dirigidas. Simulaciones (Herramientas tradicionales: canva, power point, etc) Estudio de casos: Se proporcionarán casos reales o simulados donde se presenten problemas o aplicaciones de coordinación en sistemas distribuidos. | |
| 4 | Escalabilidad en Sistemas Distribuidos. Clusters: Arquitectura de Clusters. Clústeres para computación paralela escalable. Máquinas virtuales y virtualización de clusters. Virtualización de CPU, memoria y dispositivos de E/S. | Clases magistrales con demostraciones: Exposición de conceptos y técnicas de escalabilidad; demostraciones en vivo de despliegue y balanceo de carga. Estudio de Casos: casos reales de aplicaciones distribuidas con diferentes desafíos de escalabilidad. Demostraciones prácticas. Herramientas: Herramientas de virtualización y orquestación (Docker, Kubernetes, Rabbit MQ). | 10 |
| 5 | Aplicaciones de Sistemas Distribuidos: Microkernel. Arquitectura en la nube: Virtualización de servidores. Contenedores. Optimización de servicios. Arquitectura serverless. Edge computing. Arquitectura orientada a servicios (SOA). Arquitectura de microservicios. Base de datos distribuidas. Algoritmos distribuidos. Sistemas de archivos distribuidos. | Clase magistral: Presentación de diversas aplicaciones reales de sistemas distribuidos en industrias, servicios cloud, IoT, etc. Aprendizaje basado en proyectos (ABP) Presentación: Defensa de proyecto (tipo pitch) Herramientas: Entornos para microservicios y ROA (Derivado del SOA). | 8 |

6. **Métodos de aprendizaje:** Se aplica una metodología de aprendizaje activo y experiencial centrada en la resolución de problemas de comunicación y sincronización en red. El curso combina clases magistrales con aprendizaje colaborativo mediante el desarrollo de un proyecto de programación distribuida y talleres técnicos supervisados. Se fomenta el aprendizaje autónomo a través de la investigación de protocolos y algoritmos, junto al uso de herramientas de simulación para validar la robustez y escalabilidad de los sistemas. Este enfoque se complementa con tutorías dirigidas y defensas técnicas que aseguran la integración de conceptos teóricos en soluciones de software funcionales y seguras.

7. **Métodos de evaluación:**

| Aprendizaje en contacto con el docente (62.5 %) | Aprendizaje práctico experimental (30 %) | Aprendizaje autónomo (5 %) |
|--|--|---|
| Exposiciones, Participación en clases, Debates, Exámenes escritos u orales, Talleres, Defensa de proyectos, entre otros. | Resolución de problemas prácticos, Prácticas de laboratorio, Salidas de campo o visitas técnicas, Manejo de software especializado, Prototipado técnico, Estudios de caso técnicos, entre otros. | Elaboración de informes, Resolución de problemas y ejercicios, Ensayos de investigación, Creación de mapas conceptuales, Participación en foros, entre otros. |

8. Referencias

Obligatoria:

- Coulouris, G., Dollimore, J. y Kindberg, T. (2012). *Sistemas Distribuidos Conceptos y Diseños* (5ta. Edición). Addison-Wesley.
- Tanenbaum, A. y Van Steen, M. (2023). *Sistemas Distribuidos. Principios y Paradigmas* (4ta. Edición). Pearson.

Adicional:

- Benmammam, B. (2017). *Concurrent, real-time and distributed programming in Java: Threads, RTSJ and RMI* (1a ed.). Wiley-ISTE.
- Burns, B. (2023). *Designing distributed systems: Patterns and paradigms for scalable, reliable systems using Kubernetes*. O'Reilly Media.
- Joshi, U. (2024). *Patterns of distributed systems*. Addison-Wesley.
- Kleppmann, M. (2017). *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. O'Reilly Media.
- Newman, S. (2020). *Building microservices: Designing fine-grained systems* (2a ed.). O'Reilly Media.
- Vitillo, R. (2021). *Understanding distributed systems*. Independently published.