

**Universidad Católica San Pablo (UCSP)**  
**Escuela Profesional de**  
**Ciencia de la Computación**  
**SILABO**



**CS2S1. Sistemas Operativos (Obligatorio)**

**1. Información general**

1.1 Escuela	:	Ciencia de la Computación
1.2 Curso	:	CS2S1. Sistemas Operativos
1.3 Semestre	:	6 <sup>to</sup> Semestre.
1.4 Prerrequisitos	:	CS221. Arquitectura de Computadores. (3 <sup>er</sup> Sem)
1.5 Condición	:	Obligatorio
1.6 Modalidad de aprendizaje	:	Virtual
1.7 horas	:	2 HT; 2 HP; 2 HL;
1.8 Créditos	:	4

**2. Profesores**

**3. Fundamentación del curso**

Un sistema operativo (SO) gestiona los recursos computacionales para completar la ejecución de múltiples aplicaciones y sus procesos asociados. Este curso enseña el diseño de sistemas operativos modernos; e introduce sus conceptos fundamentales que cubren la ejecución multi-programa, *scheduling*, gerencia de memoria, sistemas de archivos y seguridad. Además, el curso incluye actividades de programación en un *sistema operativo mínimo* para resolver problemas y ampliar su funcionalidad. Tenga en cuenta que estas actividades requieren mucho tiempo para completarse. Sin embargo, trabajar en ellos proporciona un valioso aprendizaje sobre los sistemas operativos.

**4. Resumen**

1. Visión general de Sistemas Operativos 2. Principios de Sistemas Operativos 3. Concurrencia 4. Planificación y despacho 5. Manejo de memoria 6. Seguridad y protección 7. Máquinas virtuales 8. Manejo de dispositivos 9. Sistema de archivos 10. Sistemas *embedded* y de tiempo real 11. Tolerancia a fallas 12. Evaluación del desempeño de sistemas

**5. Objetivos Generales**

- Estudiar el diseño de sistemas operativos modernos.
- Proveer una experiencia práctica al diseñar e implementar un sistema operativo mínimo.

**6. Contribución a los resultados (*Outcomes*)**

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (**Evaluar**)
- 4) Reconocer las responsabilidades profesionales y hacer juicios informados en el campo profesional de computación con principios éticos. (**Familiarizarse**)
- 5) Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (**Usar**)
- 6) Aplicar fundamentos de teoría de ciencias de la computación y desarrollo de software para producir soluciones basados en computación. (**Usar**)
- 7) Desarrollar tecnología computacional buscando el bien común, aportando con formación humana, capacidades científicas, tecnológicas y profesionales para solucionar problemas sociales de nuestro entorno. (**Evaluar**)

## 7. Contenido

### UNIDAD 1: Visión general de Sistemas Operativos (3)

#### Competencias:

Contenido	Objetivos Generales
<ul style="list-style-type: none"><li>• Papel y el propósito del sistema operativo.</li><li>• Funcionalidad de un sistema operativo típico.</li><li>• Los mecanismos de apoyo modelos cliente-servidor.</li><li>• Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad)</li><li>• Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.</li></ul>	<ul style="list-style-type: none"><li>• Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse]</li><li>• Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Evaluar]</li><li>• Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse]</li><li>• Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse]</li><li>• Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]</li></ul>

**Lecturas:** Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)

### UNIDAD 2: Principios de Sistemas Operativos (6)

#### Competencias:

Contenido	Objetivos Generales
<ul style="list-style-type: none"><li>• Estructuración de Sistemas Operativos (monolítico, capas, modular, los modelos micro-kernel)</li><li>• Abstracciones, procesos y recursos.</li><li>• Los conceptos de interfaces de programa de aplicación (API)</li><li>• La evolución de las técnicas de hardware / software y las necesidades de aplicación</li><li>• Organización de dispositivos.</li><li>• Interrupciones: métodos e implementaciones.</li><li>• Concepto de estado de usuario / sistema y la protección, la transición al modo kernel.</li></ul>	<ul style="list-style-type: none"><li>• Explicar el concepto de una capa lógica [Familiarizarse]</li><li>• Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarizarse]</li><li>• Describir el valor de la API y <i>middleware</i> [Familiarizarse]</li><li>• Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarizarse]</li><li>• Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Evaluar]</li><li>• Discutir las ventajas y desventajas del uso de procesamiento con interrupciones [Familiarizarse]</li><li>• Explicar el uso de una lista de dispositivos y el controlador de colas de entrada y salida [Familiarizarse]</li></ul>

**Lecturas:** Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)

UNIDAD 3: Concurrencia (9)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Diagramas de estado.</li> <li>• Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente)</li> <li>• <i>Dispatching</i> y cambio de contexto.</li> <li>• El papel de las interrupciones.</li> <li>• Gestionar el acceso a los objetos del sistema operativo de forma atómica.</li> <li>• La implementación de primitivas de sincronización.</li> <li>• Problemas de multiprocesador (spin-locks, reentrada)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarizarse]</li> <li>• Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usar]</li> <li>• Resumir el rango de mecanismos que pueden ser usados a nivel del sistema operativo para realizar sistemas concurrentes y describir los beneficios de cada uno [Familiarizarse]</li> <li>• Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarizarse]</li> <li>• Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarizarse]</li> <li>• Describir las razones para usar interrupciones, <i>dispatching</i>, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarizarse]</li> <li>• Crear diagramas de estado y transición para los problemas de dominios simples [Usar]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 4: Planificación y despacho (6)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• <i>Scheduling preemptive y non-preemptive.</i></li> <li>• <i>Scheduling</i> y políticas.</li> <li>• Procesos y subprocesos.</li> <li>• Plazos y cuestiones en tiempo real.</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar y contrastar los algoritmos comunes que se utilizan tanto para <i>scheduling preemptive</i> y <i>preemptive</i> de tareas en los sistemas operativos, como la comparación de prioridad, el rendimiento, y los esquemas de distribución equitativa [Evaluar]</li> <li>• Describir las relaciones entre los algoritmos de <i>scheduling</i> y dominios de aplicación [Familiarizarse]</li> <li>• Discutir los tipos de <i>scheduling</i> en procesadores en de corto, mediano, largo plazo y I/O [Familiarizarse]</li> <li>• Describir las diferencias entre procesos y <i>threads</i> [Familiarizarse]</li> <li>• Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar]</li> <li>• Discutir sobre la necesidad de <i>preemption</i> y <i>deadline scheduling</i> [Familiarizarse]</li> <li>• Identificar formas en que la lógica expresada en algoritmos de planificación son de aplicación a otros ámbitos, tales como I/O del disco, la programación de disco de red, programación de proyectos y problemas más allá de la computación [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 5: Manejo de memoria (6)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Revisión de la memoria física y hardware de gestión de memoria.</li> <li>• Conjuntos de trabajo y thrashing.</li> <li>• El almacenamiento en caché</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la jerarquía de la memoria y <i>tradeoffs</i> de costo-rendimiento [Familiarizarse]</li> <li>• Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarizarse]</li> <li>• Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Evaluar]</li> <li>• Describir las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarizarse]</li> <li>• Describir el motivo y el uso de memoria caché (rendimiento y proximidad, dimensión diferente de como los caches complican el aislamiento y abstracción en VM) [Familiarizarse]</li> <li>• Estudiar los conceptos de <i>thrashing</i>, tanto en términos de las razones por las que se produce y las técnicas usadas para el reconocimiento y manejo del problema [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 6: Seguridad y protección (6)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Visión general de la seguridad del sistema .</li> <li>• Política / mecanismo de separación.</li> <li>• Métodos de seguridad y dispositivos.</li> <li>• Protección, control de acceso y autenticación.</li> <li>• Las copias de seguridad.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse]</li> <li>• Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse]</li> <li>• Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse]</li> <li>• Realizar tareas de administración de sistemas sencillas de acuerdo a una política de seguridad, por ejemplo la creación de cuentas, el establecimiento de permisos, aplicación de parches y organización de backups regulares [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 7: Máquinas virtuales (6)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red)</li> <li>• Paginación y la memoria virtual.</li> <li>• Sistemas de archivos virtuales.</li> <li>• Los Hypervisores.</li> <li>• Virtualización portátil; emulación vs aislamiento.</li> <li>• Costo de la virtualización.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse]</li> <li>• Diferenciar emulación y el aislamiento [Familiarizarse]</li> <li>• Evaluar virtualización de compensaciones [Evaluar]</li> <li>• Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 8: Manejo de dispositivos (6)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Características de los dispositivos serie y paralelo.</li> <li>• Haciendo de abstracción de dispositivos.</li> <li>• Estrategias de buffering.</li> <li>• Acceso directo a memoria.</li> <li>• La recuperación de fallos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarizarse]</li> <li>• Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarizarse]</li> <li>• Explicar <i>buffering</i> y describir las estrategias para su aplicación [Familiarizarse]</li> <li>• Diferenciar los mecanismos utilizados en la interconexión de un rango de dispositivos (incluyendo dispositivos portátiles, redes, multimedia) a un ordenador y explicar las implicaciones de éstas para el diseño de un sistema operativo [Familiarizarse]</li> <li>• Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarizarse]</li> <li>• Identificar los requerimientos para recuperación de errores [Familiarizarse]</li> <li>• Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usar]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

<b>UNIDAD 9: Sistema de archivos (6)</b>	
<b>Competencias:</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial.</li> <li>• Directorios: contenido y estructura.</li> <li>• Los sistemas de archivos: partición, montar / desmontar sistemas de archivos virtuales.</li> <li>• Técnicas estándar de implementación .</li> <li>• Archivos asignados en memoria.</li> <li>• Sistemas de archivos de propósito especial.</li> <li>• Naming, búsqueda, acceso, copias de seguridad.</li> <li>• La bitacora y los sistemas de archivos estructurados (log)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarizarse]</li> <li>• Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Evaluar]</li> <li>• Resumir cómo el desarrollo de hardware ha dado lugar a cambios en las prioridades para el diseño y la gestión de sistemas de archivos [Familiarizarse]</li> <li>• Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

<b>UNIDAD 10: Sistemas <i>embedded</i> y de tiempo real (6)</b>	
<b>Competencias:</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Proceso y programación de tareas.</li> <li>• Los requisitos de gestión de memoria / disco en un entorno en tiempo real.</li> <li>• Los fracasos, los riesgos y la recuperación.</li> <li>• Preocupaciones especiales en sistemas de tiempo real.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir que hace a un sistema un sistema en tiempo real [Familiarizarse]</li> <li>• Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarizarse]</li> <li>• Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

<b>UNIDAD 11: Tolerancia a fallas (3)</b>	
<b>Competencias:</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Conceptos fundamentales: sistemas fiables y disponibles.</li> <li>• Redundancia espacial y temporal.</li> <li>• Los métodos utilizados para implementar la tolerancia a fallos.</li> <li>• Los ejemplos de los mecanismos del sistema operativo para la detección, recuperación, reinicio para implementar la tolerancia a fallos, el uso de estas técnicas para los servicios propios del sistema operativo.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarizarse]</li> <li>• Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarizarse]</li> <li>• Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

<b>UNIDAD 12: Evaluación del desempeño de sistemas (3)</b>	
<b>Competencias:</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• ¿Por qué el rendimiento del sistema debe ser evaluado?</li> <li>• ¿Qué se va a evaluar?</li> <li>• Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad.</li> <li>• Modelos de evaluación: analítica, simulación, o de implementación específico determinista.</li> <li>• Cómo recoger los datos de evaluación (perfiles y mecanismos de localización)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarizarse]</li> <li>• Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarizarse]</li> </ul>
<b>Lecturas:</b> Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

<b>8. Metodología</b>
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.</p>



## 9. Evaluar

**Evaluación Continua 1** : 20 %

**Examen parcial** : 30 %

**Evaluación Continua 2** : 20 %

**Examen final** : 30 %

## References

Anderson, Thomas and Michael Dahlin (2014). *Operating Systems: Principles and Practice*. 2nd. Recursive Books. ISBN: 978-0985673529.

Avi Silberschatz Peter Baer Galvin, Greg Gagne (2012). *Operating System Concepts, 9/E*. John Wiley & Sons, Inc. ISBN: 978-1-118-06333-0.

Stallings, William (2005). *Operating Systems: Internals and Design Principles, 5/E*. Prentice Hall. ISBN: 0-13-147954-7.

Tanenbaum, Andrew S. (2001). *Modern Operating Systems, 4/E*. Prentice Hall. ISBN: 0-13-031358-0.

Tanenbaum, Andrew S. (2006). *Operating Systems Design and Implementation, 3/E*. Prentice Hall. ISBN: 0-13-142938-8.