



Universidad de Ingeniería y Tecnología
Escuela Profesional de
Ciencia de la Computación
Silabo del curso – Periodo Académico 2018-I

1. **Código del curso y nombre:** CS2S1. Sistemas Operativos
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HP;
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E*. John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.
- [Mat99] Luis Mateu. *Apuntes de Sistemas Operativos*. Universidad de Chile, 1999.
- [Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E*. Prentice Hall, 2005. ISBN: 0-13-147954-7.
- [Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 2/E*. Prentice Hall, 2001. ISBN: 0-13-031358-0.
- [Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E*. Prentice Hall, 2006. ISBN: 0-13-142938-8.

6. Información del curso

- (a) **Breve descripción del curso** Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y la máquina.

El propósito de un sistema operativo es proveer un ambiente en que el usuario pueda ejecutar sus aplicaciones.

En este curso se estudiará el diseño del núcleo de los sistemas operativos. Además el curso contempla actividades prácticas en donde se resolverán problemas de concurrencia y se modificará el funcionamiento de un pseudo Sistema Operativo.

- (b) **Prerrequisitos:** CS221. Arquitectura de Computadores. (3^{er} Sem)

- (c) **Tipo de Curso:** Obligatorio

7. Competencias

- Conocer los elementos básicos del diseño de los sistemas operativos.

8. Contribución a los resultados (Outcomes)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Evaluar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome g**
- C6.** Capacidad para diseñar y poner en práctica las unidades estructurales mayores que utilizan algoritmos y estructuras de datos y las interfaces a través del cual estas unidades se comunican.⇒ **Outcome h**

CS8. Aplicar los principios de la interacción persona-ordenador para la evaluación y la construcción de una amplia gama de materiales, incluyendo interfaces de usuario, páginas web, sistemas multimedia y sistemas móviles.⇒ **Outcome b, g**

10. Lista de temas a estudiar en el curso

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 empujados y de tiempo real
11. Tolerancia a fallas
12. Evaluación del desempeño de sistemas

11. Metodología y Evaluación

Metodología:

Sesiones Teóricas:

El desarrollo de las sesiones teóricas está focalizado en el estudiante, a través de su participación activa, resolviendo problemas relacionados al curso con los aportes individuales y discutiendo casos reales de la industria. Los alumnos desarrollarán a lo largo del curso un proyecto de aplicación de las herramientas recibidas en una empresa.

Sesiones de Laboratorio:

Las sesiones prácticas se desarrollan en laboratorio. Las prácticas de laboratorio se realizan en equipos para fortalecer su comunicación. Al inicio de cada laboratorio se explica el desarrollo de la práctica y al término se destaca las principales conclusiones de la actividad en forma grupal.

Exposiciones individuales o grupales:

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Visión general de Sistemas Operativos (3)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse] • Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Evaluar] • Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse] • Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse] • Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse] 	<ul style="list-style-type: none"> • Papel y el propósito del sistema operativo. • Funcionalidad de un sistema operativo típico. • Los mecanismos de apoyo modelos cliente-servidor, dispositivos de mano. • Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad) • Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 2: Principios de Sistemas Operativos (6)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar el concepto de una capa lógica [Familiarizarse] • Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarizarse] • Describir el valor de la API y <i>middleware</i> [Familiarizarse] • Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarizarse] • Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Evaluar] • Discutir las ventajas y desventajas del uso de procesamiento interrumpido [Familiarizarse] • Explicar el uso de una lista de dispositivos y el controlador de colas de entrada y salida [Familiarizarse] 	<ul style="list-style-type: none"> • Métodos de estructuración (monolítico, capas, modular, los modelos micro-kernel) • Abstracciones, procesos y recursos. • Los conceptos de interfaces de programa de aplicación (API) • La evolución de las técnicas de hardware / software y las necesidades de aplicación • Organización de dispositivos. • Interrupciones: métodos e implementaciones. • Concepto de usuario de estado / sistema y la protección, la transición al modo kernel.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 3: Concurrencia (9)	
Competences esperadas: C6	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarizarse] • Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usar] • 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] • Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarizarse] • Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarizarse] • Describir las razones para usar interruptores, despacho, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarizarse] • Crear diagramas de estado y transición para los dominios de problemas simples [Usar] 	<ul style="list-style-type: none"> • Diagramas de estado. • Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente) • Despacho y cambio de contexto. • El papel de las interrupciones. • Gestionar el acceso a los objetos del sistema operativo atómica. • La implementación de primitivas de sincronización. • Cuestiones multiprocesador (spin-locks, reentrada)
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 4: Planificación y despacho (6)	
Competences esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comparar y contrastar los algoritmos comunes que se utilizan tanto para un programa preferente y no preferente de las tareas en los sistemas operativos, como la comparación de prioridad, el rendimiento, y los esquemas de distribución equitativa [Evaluar] • Describir las relaciones entre los algoritmos de planificación y dominios de aplicación [Familiarizarse] • Discutir los tipos de planeamiento de procesos <i>scheduling</i> de corto, a mediano, a largo plazo y I/O [Familiarizarse] • Describir las diferencias entre procesos y hebras [Familiarizarse] • Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar] • Hablar sobre la necesidad de tiempos límites de <i>scheduling</i> [Familiarizarse] • 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] 	<ul style="list-style-type: none"> • Planificación preventiva y no preferente. • Planificadores y políticas. • Procesos y subprocessos. • Plazos y cuestiones en tiempo real.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 5: Manejo de memoria (6)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar la jerarquía de la memoria y costo-rendimiento de intercambio [Familiarizarse] • Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarizarse] • Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Evaluar] • Defiende las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarizarse] • 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] • 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] 	<ul style="list-style-type: none"> • Revisión de la memoria física y hardware de gestión de memoria. • Conjuntos de trabajo y thrashing. • El almacenamiento en caché
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 6: Seguridad y protección (6)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Articular la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse] • Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse] • Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse] • 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] 	<ul style="list-style-type: none"> • Visión general de la seguridad del sistema . • Política / mecanismo de separación. • Métodos de seguridad y dispositivos. • Protección, control de acceso y autenticación. • Las copias de seguridad.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 7: Máquinas virtuales (6)	
Competences esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse] • Diferenciar emulación y el aislamiento [Familiarizarse] • Evaluar virtualización de compensaciones [Evaluar] • Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarizarse] 	<ul style="list-style-type: none"> • Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) • Paginación y la memoria virtual. • Sistemas de archivos virtuales. • Los Hypervisor. • Virtualización portátil; emulación vs aislamiento. • Costo de la virtualización.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 8: Manejo de dispositivos (6)	
Competences esperadas: C6	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarizarse] • Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarizarse] • Explique <i>buffering</i> y describir las estrategias para su aplicación [Familiarizarse] • 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] • Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarizarse] • Identificar los requerimientos para recuperación de errores [Familiarizarse] • Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usar] 	<ul style="list-style-type: none"> • Características de los dispositivos serie y paralelo. • Haciendo de abstracción de dispositivos. • Estrategias de buffering. • Acceso directo a memoria. • La recuperación de fallos.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 9: Sistema de archivos (6)	
Competences esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarizarse] • Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Evaluar] • 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] • Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarizarse] 	<ul style="list-style-type: none"> • Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial. • Directorios: contenido y estructura. • Los sistemas de archivos: partición, montar sistemas de archivos / desmontar, virtuales. • Técnicas estándar de implementación . • Archivos asignados en memoria. • Sistemas de archivos de propósito especial. • Naming, búsqueda, acceso, copias de seguridad. • La bitacora y los sistemas de archivos estructurados (log)
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 10: Sistemas empujados y de tiempo real (6)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir que hace a un sistema un sistema en tiempo real [Familiarizarse] • Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarizarse] • Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarizarse] 	<ul style="list-style-type: none"> • Proceso y programación de tareas. • Los requisitos de gestión de memoria / disco en un entorno en tiempo real. • Los fracasos, los riesgos y la recuperación. • Preocupaciones especiales en sistemas de tiempo real.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 11: Tolerancia a fallas (3)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarizarse] • Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarizarse] • Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarizarse] 	<ul style="list-style-type: none"> • Conceptos fundamentales: sistemas fiables y disponibles. • Redundancia espacial y temporal. • Los métodos utilizados para implementar la tolerancia a fallos. • Los ejemplos de los mecanismos del sistema operativo para la detección, recuperación, reinicie para implementar la tolerancia a fallos, el uso de estas técnicas para los servicios propios del sistema operativo.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 12: Evaluación del desempeño de sistemas (3)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarizarse] • Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarizarse] 	<ul style="list-style-type: none"> • ¿Por qué el rendimiento del sistema debe ser evaluado? • ¿Qué se va a evaluar? • Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad. • Modelos de evaluación: analítica, simulación, o de implementación específico determinista. • Cómo recoger los datos de evaluación (perfiles y mecanismos de localización)
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	



Universidad de Ingeniería y Tecnología
Escuela Profesional de
Ciencia de la Computación
Silabo del curso – Periodo Académico 2018-I

1. **Código del curso y nombre:** CS291. Ingeniería de Software I
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HP;
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Lar08] Craig Larman. *Applying UML and Patterns*. Prentice Hall, 2008.
- [Pre05] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 6th. McGraw-Hill, Mar. 2005.
- [Som08] Ian Sommerville. *Software Engineering*. 7th. ISBN: 0321210263. Addison Wesley, May 2008.

6. Información del curso

- (a) **Breve descripción del curso** La tarea de desarrollar software, excepto para aplicaciones sumamente simples, exige la ejecución de un proceso de desarrollo bien definido. Los profesionales de esta área requieren un alto grado de conocimiento de los diferentes modelos e proceso de desarrollo, para que sean capaces de elegir el más idóneo para cada proyecto de desarrollo. Por otro lado, el desarrollo de sistemas de mediana y gran escala requiere del uso de bibliotecas de patrones y componentes y del dominio de técnicas relacionadas al diseño basado en componentes.
- (b) **Prerrequisitos:** CS113. Programación Orientada a Objetos II. (3^{er} Sem) , CS271. Bases de Datos I. (4^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio

7. Competencias

- Brindar al alumno un marco teórico y práctico para el desarrollo de software bajo estándares de calidad.
- Familiarizar al alumno con los procesos de modelamiento y construcción de software a través del uso de herramientas CASE.
- Los alumnos debe ser capaces de seleccionar Arquitecturas y Plataformas tecnológicas ad-hoc a los escenarios de implementación.
- Aplicar el modelamiento basado en componentes y fin de asegurar variables como calidad, costo y *time-to-market* en los procesos de desarrollo.
- Brindar a los alumnos mejores prácticas para la verificación y validación del software.

8. Contribución a los resultados (Outcomes)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Usar**)

9. Competencias (IEEE)