LABORATORIO SISTEMAS OPERATIVOS 2 - SECCIÓN A

CÓDIGO:	0285	PUNTEO NETO LABORATORIO:	24
ESCUELA DE INGENIERÍA EN:	**CIENCIAS Y SISTEMAS**	ÁREA A LA QUE PERTENECE:	**CIENCIAS DE LA COMPUTACIÓN**
PRE REQUISITO:	281 - SISTEMAS OPERATIVOS 1	POST REQUISITO:	798 - SEMINARIO DE SISTEMAS 2 2009 - PRACTICAS FINALES INGENIERÍA CIENCIAS Y SISTEMAS
CATEGORÍA:	**OBLIGATORIO**	VIGENCIA:	**SEGUNDO SEMESTRE 2026**

Descripción del Laboratorio

El laboratorio del curso **"Sistemas Operativos 2"** está diseñado para que el estudiante aprenda a diseñar e implementar aplicaciones que hagan uso de los servicios proporcionados por el sistema operativo, así como para adquirir los conocimientos necesarios que le permitan utilizar dicho sistema a nivel de usuario avanzado.

Resumen de Ponderaciones y Tiempo de Auto-aprendizaje

TIPO	PONDERACIÓN	HORAS DE AUTO-APRENDIZAJE
Actividades en Clase	0	0
Proyectos	60	40
Prácticas	30	34
Tareas	10	5
Examen Final	0	0
TOTAL	**100**	**74**

Equipo Académico

Coordinador del Área

Nombre: **M.Sc. Luis Fernando Espino Barrios**	Correo electrónico: **usac.sistemas@gmail.com**

Docente

Nombre del Docente	Correo electrónico del Docente
Edgar Rene Ornelis Hoil	**ornelyz@ingenieria.usac.edu.gt**

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día	X		X			
Horario	7:10 - 8:50		9:00 - 10:40			
Lugar	Google Meet		Google Meet			

Tutor(es)

Tutor 1

Nombre del Tutor	**Javier Alejandro Avila Flores**	
Correo electrónico institucional	**Correo 3017077070101@ingenieria.usac.edu.gt**	

Tutor 2

Nombre del Tutor	**Alan Rodrigo Pamal De León**	
Correo electrónico institucional	**Correo 3011101580101@ingenieria.usac.edu.gt**	

Tipo		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Clase	Día					X	
	Horario					17:20 - 19:00	
	Lugar					Google Meet	
Atención al Estudiante	Día	X	X	X	X	X	
	Horario	7:00 - 20:00	7:00 - 20:00	7:00 - 20:00	7:00 - 20:00	7:00 - 20:00	
	Lugar	Virtual	Virtual	Virtual	Virtual	Virtual	

Índice

Competencias Vinculadas al Perfil del Egresado

Competencias Específicas

No.	Competencia
1	Aplica los conocimientos de su disciplina en la elaboración, fundamentación y defensa de argumentos para prevenir y resolver problemas complejos en su campo profesional, identificando y aplicando innovaciones.
2	Demuestra destreza y habilidad en la selección, uso y adaptación de herramientas metodológicas, tecnológicas, equipos especializados y en la lectura e interpretación de datos, pertinentes al contexto de su ejercicio profesional.
3	Toma decisiones profesionales con base en fundamentos teóricos, datos e información pertinente, válida y confiable.

Competencias Generales

No.	Competencia
1	Actualiza permanente sus conocimientos relacionados con TIC en general, apoyándose en las estrategias de aprendizaje apropiadas.
2	Aplica principios básicos de ingeniería, ciencias de computación y sistemas de información y comunicación, en la formulación y resolución adecuada de problemas complejos.
3	Aplica estándares de calidad, eficiencia y seguridad en la implementación adecuada de soluciones de software, hardware y TIC en general.

Competencias del Laboratorio

Competencia(s) Específica(s)

No.	Competencia	Nivel de Aprendizaje
1	El estudiante crea módulos de monitoreo del sistema mediante la lectura e interpretación de estructuras internas del kernel para extraer y exponer métricas de rendimiento como uso de CPU, memoria y procesos activos en tiempo real.	Crear
2	El estudiante desarrolla aplicaciones multihilo que gestionan de forma segura el acceso a recursos compartidos mediante la implementación de la API de PThreads y el uso de primitivas de sincronización como mutexes y semáforos para evitar condiciones de carrera, asegurar la consistencia de datos y prevenir interbloqueos (deadlocks).	Aplicar
3	El estudiante analiza la asignación de memoria física y lógica junto con el mapeo de direcciones en el sistema mediante el empleo de la MMU, tablas de páginas y algoritmos de reemplazo como FIFO o LRU para garantizar el aislamiento de procesos, prevenir la fragmentación y minimizar los fallos de página.	Analizar
4	El estudiante implementa nuevas llamadas al sistema (syscalls) y servicios personalizados dentro del núcleo, aplicando buenas prácticas de seguridad mediante la modificación del código fuente del kernel de Linux, la creación de módulos y el uso seguro de copy_from_user / copy_to_user para prevenir buffer overflows y escaladas de privilegios para extender la funcionalidad del sistema operativo permitiendo una comunicación eficiente y segura entre el espacio de usuario y el espacio de kernel.	Crear
5.	El estudiante diseña aplicaciones en espacio de usuario integrando llamadas al sistema y el sistema de archivos /proc para generar comunicación eficiente entre el kernel y los procesos de usuario expandiendo las capacidades del sistema operativo.	Crear

Competencia(s) General(es)

No.	Competencia	Nivel de Aprendizaje
1	El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	Recordar
2	El estudiante evalúa las capacidades, limitaciones y características de seguridad de distintos sistemas operativos mediante la identificación de sus fortalezas, vulnerabilidades conocidas y mecanismos de protección para seleccionarlos y aplicarlos estratégicamente, considerando criterios de	Evaluar

	seguridad, en proyectos y tareas profesionales.	

Diseño Didáctico

Sesión de Diagnóstico

Evaluación de conocimientos previos

Se aplicará una actividad diagnóstica con el objetivo de identificar el nivel de conocimientos y habilidades que los estudiantes poseen al inicio del curso. No influye en la nota final, pero es obligatoria para todos los estudiantes.

Tipo de Actividad	Descripción
Cuestionario	Se aplicará una actividad diagnóstica con el objetivo de identificar el nivel de conocimientos y habilidades que los estudiantes poseen al inicio del curso. No influye en la nota final, pero es obligatoria para todos los estudiantes.

Presentación del tutor

El tutor se presenta formalmente al grupo, compartiendo su formación académica, experiencia profesional y educativa, así como sus expectativas sobre el curso. También se abordan aspectos como normas de convivencia, canales de comunicación, disponibilidad para consultas y métodos de acompañamiento.

Presentación de los estudiantes

Se escogen un grupo de estudiantes al azar. En su presentación, se les pedirá que compartan información básica como su nombre, intereses personales o profesionales, experiencias previas relacionadas con el curso y sus expectativas. Esta actividad busca promover la interacción, el reconocimiento entre pares y la construcción de un entorno participativo y respetuoso.

Presentación del programa del curso

Se presenta el contenido del programa del curso, se aclaran dudas y se fomenta el compromiso del estudiante con su aprendizaje.

Evaluación de conocimientos del laboratorio actual

Se realiza una evaluación o práctica que permite conocer el grado de familiaridad de los estudiantes con las herramientas, entornos o competencias técnicas necesarias para el laboratorio actual.

Tipo de Actividad	Descripción
Cuestionario	Se aplicará una actividad diagnóstica con el objetivo de identificar el nivel de conocimientos y habilidades que los estudiantes poseen al inicio del curso. No

	influye en la nota final, pero es obligatoria para todos los estudiantes.

Sesión No. 2, Unidad No. 1 - Introducción a los Sistemas Operativos y el Kernel

Área Actitudinal (Saber ser)

Nombre del valor: Responsabilidad
Compromiso ético y profesional que asume el estudiante al interactuar con software de código abierto y sistemas críticos.

Área de Conocimiento (Saber)

Competencia(s)
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.

Tema	Subtema
Fundamentos de Sistemas Operativos	Gestión de recursos de hardware (CPU, Memoria, Almacenamiento).
Fundamentos de Sistemas Operativos	Tipos de sistemas operativos: por lotes, multiprogramados, multiprocesamiento, multitarea, de red y distribuidos.
Ecosistema GNU/Linux	Filosofía de software libre y código abierto.
Ecosistema GNU/Linux	Diferencia entre el kernel de Linux y el sistema operativo GNU.
Arquitectura del Kernel	Estructura del código fuente del kernel.
Arquitectura del Kernel	Escribir y compilar módulos del kernel (LKM).
Arquitectura del Kernel	Carga y descarga de módulos dinámicos.

Área de Habilidades (Saber Hacer)

Competencia
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo

analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.

Tipo de Actividad	Ponderación
Cuestionario	0

Sesión No. 3, Unidad No. 2 - Personalización del Kernel y Llamadas al Sistema

Área Actitudinal (Saber ser)

Nombre del valor: Pensamiento Critico
Capacidad de cuestionar y analizar profundamente la estructura del software base.

Área de Conocimiento (Saber)

Competencia(s)
El estudiante implementa nuevas llamadas al sistema (syscalls) y servicios personalizados dentro del núcleo mediante la modificación del código fuente del kernel de Linux y la creación de módulos para extender la funcionalidad del sistema operativo y permitir una comunicación eficiente y segura entre el espacio de usuario y el espacio de kernel.

Tema	Subtema
Gestión Avanzada del Kernel	Obtención y configuración del código fuente.
Gestión Avanzada del Kernel	Proceso de personalización, compilación e instalación.
Gestión Avanzada del Kernel	Depuración y prueba de versiones personalizadas del kernel.
Llamadas al Sistema (Syscalls)	Definición y función de la interfaz entre usuario y kernel.
Llamadas al Sistema (Syscalls)	Flujo de ejecución de una llamada al sistema.
Llamadas al Sistema (Syscalls)	Llamadas comunes para gestión de procesos: fork(), exec(), wait() y exit().
Llamadas al Sistema (Syscalls)	Metodología para el desarrollo e implementación de nuevas syscalls.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	
Tipo de Actividad	Ponderación
Cuestionario	

Sesión No. 4, Unidad No. 3 - Gestión de Procesos

Área Actitudinal (Saber ser)

Nombre del valor: Consistencia
Perseverancia en el estudio sistemático para dominar herramientas complejas.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	
El estudiante implementa nuevas llamadas al sistema (syscalls) y servicios personalizados dentro del núcleo mediante la modificación del código fuente del kernel de Linux y la creación de módulos para extender la funcionalidad del sistema operativo y permitir una comunicación eficiente y segura entre el espacio de usuario y el espacio de kernel.	
Tema	Subtema
Definición y Características de Procesos	Diferencia entre un programa y un proceso en ejecución.
Definición y Características de Procesos	Identificadores de proceso: PID, PPID, UID y GID.
Definición y Características de Procesos	Aislamiento y espacio de direcciones independiente
Estructura Interna del Proceso	Secciones de memoria: Pila (stack), Montón (heap), Código (text) y Datos (data).
Ciclo de Vida y Estados	Estados del proceso: Nuevo, Listo, Ejecución (Modo usuario/kernel), Espera, Detenido y Zombi.
Ciclo de Vida y Estados	Gestión de transiciones de estado por el sistema

	operativo.
Bloque de Control de Procesos (PCB)	La estructura task_struct en el kernel de Linux.
Bloque de Control de Procesos (PCB)	Almacenamiento de registros, información de memoria y recursos asociados.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	
Tipo de Actividad	Ponderación
Cuestionario	

Sesión No. 5, Unidad No. 4 - Hilos (Threads) y Concurrencia

Área Actitudinal (Saber ser)

Nombre del valor: Disciplina
Orden y rigor en el diseño de algoritmos.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante desarrolla aplicaciones multihilo que gestionan de forma segura el acceso a recursos compartidos mediante la implementación de la API de PThreads y el uso de primitivas de sincronización como mutexes y semáforos para evitar condiciones de carrera, asegurar la consistencia de datos y prevenir interbloqueos (deadlocks).	
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	
Tema	Subtema
Fundamentos de Hilos	Definición de hilos como unidades mínimas de ejecución.
Fundamentos de Hilos	Compartición de recursos dentro de un mismo proceso (memoria y archivos).
Ejecución Multitarea	Concurrencia: Intercalación de tareas para simular simultaneidad.

Ejecución Multitarea	Paralelismo: Ejecución física simultánea en múltiples núcleos o procesadores.
Implementación y Gestión de Hilos	Creación, gestión y terminación de subprocesos.
Implementación y Gestión de Hilos	Uso de la biblioteca estándar PThreads.
Implementación y Gestión de Hilos	Tipos de hilos: Nivel de Usuario vs. Nivel de Kernel.
Implementación y Gestión de Hilos	Consideraciones de diseño: compatibilidad y eficiencia.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante desarrolla aplicaciones multihilo que gestionan de forma segura el acceso a recursos compartidos mediante la implementación de la API de PThreads y el uso de primitivas de sincronización como mutexes y semáforos para evitar condiciones de carrera, asegurar la consistencia de datos y prevenir interbloqueos (deadlocks).	
Tipo de Actividad	Ponderación
Cuestionario	

Sesión No. 6, Unidad No. 5 - Sincronización de Procesos

Área Actitudinal (Saber ser)

Nombre del valor: Criterio y toma de decisiones
Evaluación responsable de alternativas técnicas.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante desarrolla aplicaciones multihilo que gestionan de forma segura el acceso a recursos compartidos mediante la implementación de la API de PThreads y el uso de primitivas de sincronización como mutexes y semáforos para evitar condiciones de carrera, asegurar la consistencia de datos y prevenir interbloqueos (deadlocks).	
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	
Tema	Subtema
Cooperación entre Procesos	Problemas derivados del acceso a recursos compartidos.

Cooperación entre Procesos	Condiciones de carrera (Race Conditions).
La Sección Crítica	Identificación de segmentos de código sensibles.
La Sección Crítica	Requisitos para la solución de exclusión mutua: Progreso y espera limitada.
Mecanismos de Sincronización	Semáforos: Operaciones atómicas wait y signal.
Mecanismos de Sincronización	Mutexes: Bloqueos binarios para la exclusión mutua.
Gestión de Fallos de Sincronización	Detección de interbloqueos (Deadlocks).
Gestión de Fallos de Sincronización	Estrategias de prevención y recuperación de errores en sistemas concurrentes.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante desarrolla aplicaciones multihilo que gestionan de forma segura el acceso a recursos compartidos mediante la implementación de la API de PThreads y el uso de primitivas de sincronización como mutexes y semáforos para evitar condiciones de carrera, asegurar la consistencia de datos y prevenir interbloqueos (deadlocks).	
Tipo de Actividad	Ponderación
Cuestionario	

Sesión No. 7, Unidad No. 6 - Gestión de Memoria

Área Actitudinal (Saber ser)

Nombre del valor: Equidad
Mantener coherencia entre los principios teóricos de gestión de memoria y su aplicación práctica en el desarrollo de software.

Área de Conocimiento (Saber)

Competencia(s)
El estudiante analiza la asignación de memoria física y lógica junto con el mapeo de direcciones en el sistema mediante el empleo de la MMU, tablas de páginas y algoritmos de reemplazo como FIFO o LRU para garantizar el aislamiento de procesos, prevenir la fragmentación y minimizar los fallos de página.
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.

Tema	Subtema
Memoria Virtual	Algoritmos de reemplazo de páginas: FIFO, LRU y Óptimo — análisis comparativo.
Memoria Virtual	Paginación bajo demanda y manejo de fallos de página (page fault).
Fragmentación	Fragmentación interna y externa: impacto en rendimiento y estrategias de mitigación.
Traducción de Direcciones	MMU y tablas de páginas multinivel: traducción de direcciones virtuales a físicas.
Aislamiento de Procesos	Aislamiento mediante espacios de direcciones virtuales independientes.
Modelos de Memoria	Segmentación vs. paginación: comparativa y uso en sistemas modernos.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante analiza la asignación de memoria física y lógica junto con el mapeo de direcciones en el sistema mediante el empleo de la MMU, tablas de páginas y algoritmos de reemplazo como FIFO o LRU para garantizar el aislamiento de procesos, prevenir la fragmentación y minimizar los fallos de página.	
Tipo de Actividad	Ponderación
Cuestionario	

Sesión No. 8, Unidad No. 7 - Monitoreo del Sistema

Área Actitudinal (Saber ser)

Nombre del valor: Precaucion
Adoptar medidas preventivas para reducir riesgos al interactuar con estructuras internas del kernel.

Área de Conocimiento (Saber)

Competencia(s)
El estudiante crea módulos de monitoreo del sistema mediante la lectura e interpretación de estructuras internas del kernel para extraer y exponer métricas de rendimiento como uso de CPU, memoria y procesos

activos en tiempo real	
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	
Tema	Subtema
Introducción al Monitoreo	Métricas clave de rendimiento: CPU, memoria, procesos.
Métricas de CPU	Estructuras internas del kernel: jiffies y kstat para obtener uso de CPU.
Métricas de Memoria	Extracción de uso de memoria: MemTotal, MemFree, Buffers y Cached.
Listado de Procesos	Iteración de procesos activos con for_each_process y task_struct.
Sistema de Archivos /proc	Exportación de métricas del kernel a través del sistema de archivos virtual /proc.
Comparativa de Interfaces	Syscalls personalizadas vs. /proc: ventajas, limitaciones y casos de uso.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante crea módulos de monitoreo del sistema mediante la lectura e interpretación de estructuras internas del kernel para extraer y exponer métricas de rendimiento como uso de CPU, memoria y procesos activos en tiempo real	
Tipo de Actividad	Ponderación
Cuestionario	

Sesión No. 9, Unidad No. 8 - Comunicación Kernel-Usuario

Área Actitudinal (Saber ser)

Nombre del valor: Adaptabilidad
Ajustarse con flexibilidad a nuevas circunstancias, cambios o desafíos sin perder efectividad.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante diseña aplicaciones en espacio de usuario integrando llamadas al sistema y el sistema de archivos /proc para generar comunicación eficiente entre el kernel y los procesos de usuario expandiendo las capacidades del sistema operativo	
El estudiante crea módulos de monitoreo del sistema mediante la lectura e interpretación de estructuras internas del kernel para extraer y exponer métricas de rendimiento como uso de CPU, memoria y procesos activos en tiempo real.	
Tema	**Subtema**
Sistema de Archivos /proc	Fundamentos, estructura interna y navegación del sistema /proc.
Entradas /proc Personalizadas	Creación de entradas /proc con la seq_file API.
Comunicación Bidireccional	Lectura y escritura en /proc para comunicación bidireccional con el kernel.
Netlink Sockets	Comunicación asíncrona entre el kernel y el espacio de usuario con Netlink.
Diseño de Aplicaciones	Diseño de un daemon de usuario que consuma métricas del kernel en tiempo real.
Integración de Fuentes	Integración de múltiples fuentes de datos del kernel en una aplicación de usuario.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante diseña aplicaciones en espacio de usuario integrando llamadas al sistema y el sistema de archivos /proc para generar comunicación eficiente entre el kernel y los procesos de usuario expandiendo las capacidades del sistema operativo	
Tipo de Actividad	**Ponderación**
Cuestionario	

Sesión No. 10, Unidad No. 9 - Seguridad en el Kernel y Syscalls

Área Actitudinal (Saber ser)

Nombre del valor: Ética
Guiarse por principios morales que orientan el comportamiento hacia lo correcto, honesto y justo.

Área de Conocimiento (Saber)

Competencia(s)	
El estudiante implementa nuevas llamadas al sistema (syscalls) y servicios personalizados dentro del núcleo, aplicando buenas prácticas de seguridad mediante la modificación del código fuente del kernel de Linux, la creación de módulos y el uso seguro de copy_from_user / copy_to_user para prevenir buffer overflows y escaladas de privilegios para extender la funcionalidad del sistema operativo permitiendo una comunicación eficiente y segura entre el espacio de usuario y el espacio de kernel.	
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.	
Tema	Subtema
Superficie de Ataque	Vectores comunes y vulnerabilidades históricas del kernel (CVEs relevantes).
Buffer Overflow en Kernel	Análisis, demostración controlada y técnicas de mitigación de buffer overflows en módulos.
Escalada de Privilegios	Explotación de privilege escalation desde espacio de usuario y contramedidas en el kernel.
Desarrollo Seguro de Syscalls	Uso obligatorio de copy_from_user / copy_to_user y validación estricta de parámetros.
Mecanismos de Protección del SO	SELinux, AppArmor, cgroups y namespaces: configuración y casos de uso.
Auditoría del Kernel	Detección de anomalías y trazado de syscalls con auditd y eBPF.

Área de Habilidades (Saber Hacer)

Competencia
El estudiante implementa nuevas llamadas al sistema (syscalls) y servicios personalizados dentro del núcleo, aplicando buenas prácticas de seguridad mediante la modificación del código fuente del kernel de Linux, la creación de módulos y el uso seguro de copy_from_user / copy_to_user para prevenir buffer overflows y escaladas de privilegios para extender la funcionalidad del sistema operativo permitiendo una comunicación eficiente y segura entre el espacio de usuario y el espacio de kernel.

Tipo de Actividad	Ponderación
Cuestionario	

Sesión No. 11, Unidad No. 10 - Evaluación y Tendencias de Sistemas Operativos

Área Actitudinal (Saber ser)

Nombre del valor: Trabajo en Equipo
Colaborar y coordinarse con otros para alcanzar objetivos comunes, aportando habilidades individuales en beneficio del grupo.

Área de Conocimiento (Saber)

Competencia(s)
El estudiante evalúa las capacidades, limitaciones y características de seguridad de distintos sistemas operativos mediante la identificación de sus fortalezas, vulnerabilidades conocidas y mecanismos de protección (SELinux, AppArmor, namespaces) para seleccionarlos y aplicarlos estratégicamente, considerando criterios de seguridad, en proyectos y tareas profesionales.
El estudiante describe la arquitectura y los componentes fundamentales de un Sistema Operativo analizando la separación entre el espacio de usuario y el espacio de kernel para establecer las bases conceptuales necesarias para la modificación del sistema.

Tema	Subtema
Arquitecturas de Kernel	Comparativa: monolítica, microkernel e híbrida — pros, contras y ejemplos.
Casos de Uso Reales	Linux, Windows NT y macOS XNU: fortalezas, limitaciones y postura de seguridad.
Sistemas en Tiempo Real	RTOS: características, ventajas y aplicaciones industriales y embebidas.
Virtualización Ligera	Contenedores (Docker, LXC) e impacto en el rol del SO moderno.
Tendencias Emergentes	Unikernels, SO en la nube y SO seguros por diseño

	(Qubes OS, seL4).
Criterios de Selección	Rendimiento, seguridad y mantenibilidad como ejes para elegir un SO en proyectos de ingeniería.

Área de Habilidades (Saber Hacer)

Competencia	
El estudiante evalúa las capacidades, limitaciones y características de seguridad de distintos sistemas operativos mediante la identificación de sus fortalezas, vulnerabilidades conocidas y mecanismos de protección para seleccionarlos y aplicarlos estratégicamente, considerando criterios de seguridad, en proyectos y tareas profesionales.	
Tipo de Actividad	Ponderación
Cuestionario	

Rúbrica de Evaluación

Cada una de las actividades del laboratorio (proyectos, prácticas, tareas y otras) cuenta con una rúbrica de evaluación específica, la cual está detallada en el documento que se entrega al estudiante al momento de asignar la actividad. Estas rúbricas describen los criterios de evaluación, niveles de desempeño esperados y la ponderación correspondiente de cada aspecto evaluado.

Es **responsabilidad del estudiante** leer detenidamente la rúbrica asignada antes de iniciar el desarrollo de la actividad. Comprender los criterios de evaluación no solo permite orientar adecuadamente el trabajo, sino también mejorar el desempeño académico y fomentar la autorregulación del aprendizaje.

En caso de no recibir la rúbrica al momento de la asignación, el estudiante **debe solicitarla directamente al tutor académico**, ya que constituye una herramienta esencial para el cumplimiento de los objetivos de aprendizaje y la evaluación transparente.

Normativa Académica y Ética del Curso

En concordancia con el perfil del estudiante de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, se espera un alto nivel de compromiso con la excelencia académica y la ética profesional. Por ello, que se establece los siguientes lineamientos de carácter obligatorio que regulan el comportamiento académico del estudiante:

Plagio y copias

- Todo proyecto será sometido a verificación para confirmar su autoría y originalidad, con la finalidad de evitar cualquier plagio, copia o que la actividad no haya sido realizada por el estudiante.
- Cualquier evidencia de lo antes descrito en las distintas actividades será sancionada con una calificación de 0 (cero) y el caso será reportado al Docente quien a su vez informará a la Escuela de Ciencias y Sistemas para su seguimiento institucional.

Prórrogas y reposiciones

- No se otorgarán prórrogas para entregas de actividades.
- No se permitirá la reposición de proyectos bajo ninguna circunstancia.

Requisitos para evaluación final del curso

- Es obligatorio aprobar el laboratorio para tener derecho a la evaluación final del curso.
- La calificación de prácticas, proyectos y otras actividades que se indique será asignada de forma presencial, en la fecha y hora establecidas por el tutor académico.

Asistencia

- Para obtener la nota del laboratorio, se requiere un mínimo del 80% de asistencia a las sesiones de laboratorio.
- En caso de inasistencia, sólo se aceptarán justificaciones válidas respaldadas por constancia oficial.

Entregas

- No se aceptarán entregas tardías de tareas, prácticas, exámenes cortos, exámenes finales o proyectos sin justificación.

Medio oficial de entrega

- La plataforma UEDI de la Facultad será el único medio oficial para la entrega de actividades del curso.

Bibliografía

- A. Silberschatz, P. Galvin, G. Gagne, Wiley (2005). Operating System Concepts. Séptima edición
- J. Carretero, P. De Miguel, F. García, F. Pérez, Mc Graw Hill (2001). Sistemas Operativos, Una Visión Aplicada. Primera edición.
- Andrew S. Tanenbaum, Prentice Hall (2003). Sistemas Operativos Modernos. Segunda edición.