

FORMATO 1

PLANIFICACIÓN DE CURSO

Segundo Semestre académico 2021 - Docencia Remota

I. ACTIVIDAD CURRICULAR Y CARGA HORARIA

Asignatura:	Programación de Software de Sistemas	Código:	COM2002
Semestre de la Carrera:	IV Semestre		
Carrera:	Ingeniería Civil en Computación		
Escuela:	Escuela de Ingeniería		
Docente(s):	Diego Madariaga		
Ayudante(s):	Lucas Torrealba		
Horario:	Cátedra: Lunes 12:00 - 13:30, Viernes 12:00 - 13:30 Ayudantía: Jueves 08:30 - 10:00		

Créditos SCT:	6
Carga horaria semestral ¹ :	180 horas
Carga horaria semanal:	12 horas

Tiempo de trabajo sincrónico semanal:	4,5 horas
Tiempo de trabajo asincrónico semanal:	7,5 horas

II. RESULTADOS U OBJETIVOS DE APRENDIZAJE ESPERADOS ESTE SEMESTRE

1)	Escriba y entienda programas eficientes en lenguaje de programación C, utilizando las funciones básicas que provee el Sistema Operativo Linux tanto para la administración de sistemas, para hacer programas eficientes y para programación concurrente.
2)	Maneje los conceptos básicos de la programación de sistemas: arquitectura de computadores, direcciones de memoria, notación hexadecimal, representación de enteros, bits.
3)	Desarrolle software de sistemas para Linux usando las funciones de manejo de memoria, Entrada/Salida, sistema de archivos, sockets y threads.

¹ Considere que 1 crédito SCT equivale a 30 horas de trabajo total (presencial/sincrónico y autónomo/asincrónico) en el semestre.

III. UNIDADES, CONTENIDOS Y ACTIVIDADES

UNIDAD: Programación eficiente en C				
Semana	Contenidos	Actividades de enseñanza y aprendizaje		Actividades de evaluación diagnóstica, formativa y/o sumativa
		Tiempo sincrónico	Tiempo asincrónico (trabajo autónomo del o la estudiante)	
1	Introducción al Lenguaje C	4,5	7,5	
2	Representación de datos y manejo de memoria	4,5	7,5	Lunes: Publicación Tarea 1 (partes a y b)
3	Arreglos y punteros	4,5	7,5	Miércoles: Entrega Tarea 1 parte a
4	Strings en C	4,5	7,5	
5	Estructuras	4,5	7,5	Lunes: Entrega Tarea 1 parte b

UNIDAD: Programación concurrente				
Semana	Contenidos	Actividades de enseñanza y aprendizaje		Actividades de evaluación diagnóstica, formativa y/o sumativa
		Tiempo sincrónico	Tiempo asincrónico (trabajo autónomo del o la estudiante)	
6	Threads y C (pthreads)	4,5	7,5	Lunes: Publicación Tarea 2 (partes a y b)

7	Exclusión mutua y herramientas de sincronización (I)	4,5	7,5	Miércoles: Entrega Tarea 2 parte a
8	Exclusión mutua y herramientas de sincronización (II)	4,5	7,5	
9	Problemas clásicos de sincronización (I)	4,5	7,5	Lunes: Entrega Tarea 2 parte b
10	Problemas clásicos de sincronización (II)	4,5	7,5	

UNIDAD: Programación de procesos				
Semana	Contenidos	Actividades de enseñanza y aprendizaje		Actividades de evaluación diagnóstica, formativa y/o sumativa
		Tiempo sincrónico	Tiempo asincrónico (trabajo autónomo del o la estudiante)	
11	Entrada y salida	4,5	7,5	Lunes: Publicación Tarea 3 (partes a y b)
12	Procesos UNIX (I)	4,5	7,5	Miércoles: Entrega Tarea 3 parte a
13	Procesos UNIX (II)	4,5	7,5	
14	Manejo de señales	4,5	7,5	Jueves: Entrega Tarea 3 parte b

IV. CONDICIONES Y POLÍTICAS DE EVALUACIÓN

La evaluación del curso contará de 3 tareas computacionales, las cuales evaluarán cada una de las 3 unidades del curso. Cada tarea contará de dos partes (a y b), con el fin de propiciar un trabajo continuo. Ambas partes serán publicadas en conjunto, pero tendrán distintas fechas de entrega. La primera parte de cada tarea contará con ~10 días para su entrega, mientras que la segunda parte contará con ~21 días para su entrega. La primera parte de cada tarea evaluará los contenidos iniciales de cada unidad y tendrá una ponderación menor en la nota final de la tarea (20-30%). Se espera que la parte a sea resuelta en un tiempo significativamente menor que la parte b. La segunda parte de cada tarea será una extensión de la primera parte y evaluará los contenidos fundamentales de cada unidad.

Adicionalmente, se publicarán ejercicios de programación semanales, de carácter opcional. Estos ejercicios están planificados para ser realizados en un tiempo de 20 a 30 minutos. En total, se contará con 13 ejercicios durante el semestre. Estos ejercicios serán de ayuda para poner en práctica continuamente los contenidos vistos en cada semana. La resolución de cada ejercicio se traducirá en una bonificación de 1 décima en la nota de cualquiera de las 3 tareas de programación.

Se encuentra estrictamente prohibido compartir cualquier porción de código. Las entregas serán comparadas exhaustivamente en busca de plagio.

El curso no cuenta con examen. El curso será aprobado si el promedio de notas de las tareas es igual o superior a 4,0.

V. BIBLIOGRAFÍA Y RECURSOS OBLIGATORIOS

Brian Kernighan, Dennis Ritchie. The C Programming Language. Prentice Hall. 2nd Edition, 1988.

BIBLIOGRAFÍA Y RECURSOS COMPLEMENTARIOS