UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA
VICE RECTORADO ACADÉMICO
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS
DEPARTAMENTO ACADÉMICO DE Ingeniería de Sistemas e Informática
SÍLABO 2018-A

ASIGNATURA: PROGRAMACIÓN DE SISTEMAS

1. INFORMACIÓN ACADÉMICA

Periodo académico:	2018-A	
Escuela profesional:	Ingeniería de Sistemas	
Código de la asignatura :	1303126	
Nombre de la asignatura:	Programación de Sistemas	
Semestre:	V (Quinto)	
Características:	Semestral	
Duración:	17 Semanas	
Número de horas (Semestral)	Teóricas:	2
	Prácticas:	0
	Laboratorio:	4
Número de créditos:	4 (Cuatro)	
Prerrequisitos:	1302220 – Arquitectura de Computadores	

2. INFORMACIÓN ADMINISTRATIVA

Docente	**Grado Académico**	**Dpto. Académico**	**Total de horas**	**Horario / Actividad**
Alfredo Paz Valderrama	Ingeniero de Sistemas	Ingeniería de Sistemas e Informática	6horas	Martes 15:40-17:20 Laboratorio (Lab-EPIS); Miércoles 08:40-10:20 Laboratorio (Lab-EPIS); Miércoles 10:20-12:00 Teoría (Aula 302)

3. FUNDAMENTACIÓN (JUSTIFICACIÓN)

"La ingeniería del software, los algoritmos y estructuras de datos son muy importantes: nos permiten hacer abstracciones, reusar y mantener código además de evaluar la eficiencia de los programas; sin embargo, esto puede no ser suficiente, las constantes del tiempo asintótico importan en el mundo real. Este curso permitirá entender y conocer el proceso interno de la ejecución de los programas: se explicará cómo es que los códigos fuente, traducidos a código de máquina se ejecutan a bajo nivel. Con la comprensión de esto, se podrán corregir errores que programación que afectan la seguridad y confiabilidad de los sistemas. Este curso podría ser visto como un curso para Hackers, en el sentido de que los programadores entiendan lo que ocurre detrás de bambalinas."

Los participantes del curso podrán mejorar sus habilidades de programación a bajo nivel. El curso hará uso del lenguaje de programación C dónde se estudiarán conceptos tales como: manipulicación de la memoria (punteros) y llamadas al sistema, programación a bajo nivel, procesos pesados (*fork*), señales, programación *Bash* y llamadas al sistema.

4. COMPETENCIAS DEL CURSO

a) Comprende la representación de números en el computador, teniendo en cuenta ambientes de 32 y 64 bits y distintos tipos de datos como enteros y reales, con y sin signo, permitiendéndole encontrar, explicar y resolver problemas que afectan la ejecución correcta y segura de sistemas.

b) Valora como los detalles de programación a bajo nivel pueden afectar el desempeño de los sistemas en el mundo real.

c) Programa aplicaciones con múltiples hilos de ejecución en la máquina local y en máquinas remotas, entendiendo su necesidad y utilidad para el uso óptimo de los recursos en un contexto de multitarea.

5. CONTENIDOS

Unidad 1: Introducción

1. Motivación, explicación del sílabo, reglas del curso.
2. Un entorno de programación tipo UNIX: programación en bash, variables condicionales, ciclos.
3. El lenguaje de programación C: El entorno de programación, historia, tipos de datos, estructuras, punteros.
4. Compilación separada, Makefile

   Lectura Obligatoria [KR98]
   Lectura Sugerida [Liba, Libb, Libc]

Unidad 2: Datos y memoria

1. Representación de Datos: Almacenamiento de la información, representación de enteros, aritmética de enteros, punto flotante.
2. Representación de Programas a nivel de máquina: Codificación de programas, formato de datos, acceso a la información, operaciones lógicas y aritméticas, estructuras de control.
3. Jerarquías de Memoria: Tecnologías de almacenamiento, Localidad, Jerarquías de memoria, memoria cache.

   Lectura Obligatoria [BO15]
   Lectura Sugerida [KR98, Liba, Pet00]

Unidad 3: Ejecución de programas

1. Enlace: Directores de compilación, enlace estático, archivos objeto, archivos objeto reubicables, símbolos y tablas de símbolos, resolución de símbolos, reubicación, archivos objeto ejecutables, enlace dinámico con bibliotecas compartidas, cargando y enlazando bibliotecas desde aplicaciones, código independiente de la posición, herramientas para manipular archivos objeto.
2. Control de flujo excepcional: Excepciones, Procesos, Llamadas al sistema para el manejo de errores, Control de procesos, Señales, saltos no locales, herramientas.
3. Entrada/Salida a nivel del sistema: Entrada/Salida en UNIX, apertura y cierre de archivos, escribiendo y leyendo archivos.

   Lectura Obligatoria [BO15]
   Lectura Sugerida [KR98, Liba]

Unidad 4: Redes y concurrencia

1. Programación de Redes: El modelo programación cliente/servidor, Redes, La internet IP global, la interfaz socket, servidores web.
2. Programación Concurrente: Programación concurrente con procesos, programación concurrente con E/S multiplexada, programación concurrente con threads.

   Lectura Obligatoria [BO15, APSI16]
   Lectura Sugerida [KR98, Liba]

6. ESTRATEGIAS DE ENSEÑANZA

1. **Métodos:**

   - Método expositivo en varias clases teóricas.
   - Método basado en problemas
   - Método basado en proyectos

2. **Medios:**

- El curso cuenta con una página web:
  `https://sites.google.com/a/episunsa.edu.pe/systemprogramming`
- también se cuenta con una lista de discusión:
  `https://groups.google.com/forum/#!forum/ps-episunsa`
- se usarán las diapositivas oficiales del libro del curso
- el software utilizado será distribuido en clases de laboratorio y usará la gamificación para el desarrollo de las tareas.

3. **Formas de organización:**

   A) CLASES TEÓRICAS: En el curso tendrá una clase introductoria y varias clases magistrales, durante las clases se insentivirá la participación de los alumnos haciendo preguntas que cuestionen su conocimiento sobre los temas tratados en clase.

   B) SEMINARIOS: Los alumnos también expondrán temas de investigación propuestos por el profesor.

   C) PRÁCTICAS: Los alumnos resolverán ejercicios en clases, esto les permitirá aplicar los conceptos teóricos y darles un sentido de utilidad.

   D) LABORATORIO: Estas clases se realizarán en grupos de no más de 20 personas, en los laboratorios de la Escuela Prorfesional de Ingeniería de Sistemas.

4. **Actividades:**
   El curso contará con varias tareas y un trabajo final que implique *investigación formativa*

5. **Seguimiento:**
   Se usará un sistema de control de versiones para el seguimiento de trabajos y tareas, esto permitirá evaluar la dedicación y tiempo dedicados a las tareas y trabajos, así como evitar que haya copias.

7. CRONOGRAMA ACADÉMICO

Semana	Tema/Evaluación	Docente	Avance
1	Tema 1	Alfredo Paz	6 %
2	Tema 1	Alfredo Paz	12 %
3	Tema 1	Alfredo Paz	18 %
4	Tema 1	Alfredo Paz	24 %
5	Tema 2	Alfredo Paz	29 %
6	Tema 2	Alfredo Paz	35 %
7	Tema 2	Alfredo Paz	41 %
8	Tema 2, Examen	Alfredo Paz	47 %
9	Tema 3	Alfredo Paz	53 %
10	Tema 3	Alfredo Paz	59 %
11	Tema 3	Alfredo Paz	65 %
12	Tema 4	Alfredo Paz	71 %
13	Tema 4	Alfredo Paz	76 %
14	Tema 4	Alfredo Paz	82 %
15	Tema 4	Alfredo Paz	88 %
16	Tema 4, sustitutorio, trabajo final	Alfredo Paz	94 %
17	Examen	Alfredo Paz	100 %

8. ESTRATEGIAS DE EVALUACIÓN

Evaluación del Aprendizaje

La evaluación tendrá los siguientes componentes:

1. **Evaluación Continua.**
   a) (NT) Nota de Trabajos e Intervencion en clase y laboratorios 40 %
   b) (NP) Nota del proyecto de curso 15 %
2. **Evaluación Periódica**
   (NE) Nota de Examenes 45 %, esta nota se divide en:
   a) examen de Medio Semestre 40 %,

b) examen Final 60 %.

3. **Examen Subsanación o Recuperación (Sustitutorio):**
   Los alumnos que deseen podrán rendir una evaluación adicional que reemplazará al examen de medio semestre, dicha evaluación se dará sin apuntes.

Los exámenes de medio semestre y final se tomarán con apuntes.

9. REQUISITOS DE APROBACIÓN DE LA ASIGNATURA

La nota final (NF) se obtiene de la siguiente manera:

$$NF = 0,45 * NE + 0,40 * NT + 0,15 * NP$$

Los alumnos tendrán la oportunidad de rezagar un examen parcial con un plazo de 72 horas y por causas debidamente justificadas y autorizadas por la dirección de la escuela.

Posterior a la aplicación de una prueba se realizan las siguientes actividades:

- publicación del patrón de respuestas,
- solución de las preguntas del examen,
- acceso de la prueba por parte del estudiante,
- recalificación cuando es pertinente,
- publicación de los resultados usando software personalizado;
- después de todas estas actividades la nota es inmodificable.

Las calificaciones se registran en la web según cronograma.

Los exámenes son acumulativos.

Las tareas de programación deben estar correctamente indentadas, caso contrario, el profesor podrá restar puntos o incluso califcar la tarea con nota cero.

Nota sobre honestidad

La honestidad será un factor determinante en la evaluación: Los alumnos que tengan actitudes deshonesta en alguna de sus tareas, trabajos o exámen tendrán nota 0.

Actos que se consideran deshonestos

Copiar la solución de otro durante el exámen Esto incluye mirar al compañero o usar medio electrónicos (celular, etc.)

Compartir código fuente Copiar, cambiar de nombre a las variables, mostrar el código a un compañero, descargar el código de Internet, explicar el código a un compañero. Tener cuidado de no dejar copias de las tareas en lugares públicos.

Consultoría Recibir ayuda en la solución de la tarea, esta puede ser en persona, por un compañero de años superiores, por foros de discusión en Internet, etc.

Realizar los trabajos individuales en grupo Las tareas pueden tener soluciones diversas, si estas son individuales no deben reunirse para hacerlas.

Realizar las tareas grupales de manera individual Qué sólo un compañero haga toda la tarea del grupo, que cada integrante del grupo haga una parte de la tarea, pero que no tenga idea de las demás partes. Las tareas en grupo deben ser hechas en grupo, por lo que se requiere coordinación, no sólo en la distribución del trabajo, sino en la solución de los problemas que se puedan presentar. El grupo debe trabajar como un equipo.

Actos que NO se consideran como deshonestos

Explicar lo que se pide en la tarea Se puede pedir ayuda al profesor o los compañeros para entender lo que se pide en la tarea, pero siendo cuidadosos de no explicar la solución, sólo el enunciado de lo que se pide.

Explicar los temas o conceptos Si algún tema o concepto no se entiende, fuera del horario de clase, se puede pedir al profesor o algún compañero ayuda.

Llevar apuntes Se pueden llevar apuntes a los exámenes y a las evaluaciones en los laboratorios, estos apuntes podrán ayudar a recordar comandos, códigos, etc.

10. BIBLIOGRAFÍA

Obligatoria

[APSI16] Natela Archvadze, Merab Pkhovelishvili, Lia Shetsiruli, and Otar Ioseliani. The modern approaches in parallel programming. *Computer Science And Telecommunications*, 49(3):30 – 33, 2016.

[BO15] Randal E. Bryant and David R. O'Hallaron. *Computer Systems: A Programmer's Perspective*. Pearson, 3rd edition, 2015.

[KR98] Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language, Second Edition*. Prentice Hall, 1998.

Sugerida

[KR98] Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language, Second Edition*. Prentice Hall, 1998.

[Liba] Stanford CS Education Library. Essential c. `http://cslibrary.stanford.edu/101/EssentialC.pdf`. última fecha de acceso: 11 de enero del 2018.

[Libb] Stanford CS Education Library. Pointers and memory. `http://cslibrary.stanford.edu/102/PointersAndMemory.pdf`. última fecha de acceso: 11 de enero del 2018.

[Libc] Stanford CS Education Library. Unix programming tools. `http://cslibrary.stanford.edu/107/UnixProgrammingTools.pdf`. última fecha de acceso: 11 de enero del 2018.

[Pet00] C. Petzold. *Code: the hidden language of computer hardware and software*. DV-Undefined. Microsoft Press, 2000.

Arequipa, enero del 2018

Alfredo Paz Valderrama