



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE

SÍLABO

1. INFORMACIÓN GENERAL

- I.1. Nombre y código de la asignatura : ALGORÍTMICA I – 20W0301
- I.2. Plan Curricular : 2018
- I.3. Número de créditos : 04
- I.4. Número de horas semanales : Teoría: 03 horas, Laboratorio: 02 horas
- I.5. Ciclo de estudio : III
- I.6. Periodo Académico : 2021 - I
- I.7. Pre-requisitos : ---
- I.8. Profesores : Gustavo Arredondo Castillo
garredondoc@unmsm.edu.pe
Lazaro Florian Mota Alva
lmotaa@unmsm.edu.pe
Hugo David Calderón Vilca
hcalderonv@unmsm.edu.pe

2. SUMILLA

Esta asignatura de naturaleza teórico practica del área de especialidad, pretende que el estudiante desarrolle productos de software, de manera eficiente y efectiva, a partir del conocimiento de métodos, técnicas y herramientas, basados en estándares internacionales de calidad, para el desarrollo de software, con actitud creativa y responsable. Se desarrollan inicialmente las estructuras de control secuenciales, condicionales y repetitivas, luego la programación modular con subprogramas, también presentar los conceptos sobre punteros, arreglos, registros y finalmente archivos.

3. COMPETENCIAS GENERALES:

Al finalizar la asignatura, el estudiante tendrá las siguientes competencias generales:

- CG01: Gestiona la información y la difusión de conocimientos con adecuada comunicación oral y escrita de la propia profesión, ejerciendo el derecho de libertad de pensamiento con responsabilidad
- CG02: Capacidad de análisis y síntesis en la toma de decisiones con responsabilidad, sentido crítico y autocrítico
- CG03: Desempeña su profesión con liderazgo, adecuándose a los cambios y a las nuevas tendencias, comprometido con la paz, medio ambiente, equidad de género, defensa de los derechos humanos y valores democráticos
- CG04: Trabaja en equipo con una perspectiva transdisciplinar para comprender y transformar la realidad compleja
- CG05: Genera nuevos conocimientos que aportan al desarrollo de la sociedad mediante la investigación, con sentido ético
- CG06: Aplica conocimientos a la práctica para resolver problemas con compromiso ético

COMPETENCIAS ESPECÍFICAS

Al finalizar la asignatura, el estudiante tendrá las siguientes competencias específicas:

- CE2 Capacidad de Análisis
- CE3 Pensamiento Crítico
- CE4 Comunicación oral y escrita
- CE11: Diseña, implementa, verifica y valida pruebas de las soluciones de software
- CE14 Aplica metodologías, métodos Técnicas

4. PROGRAMACIÓN

UNIDAD DIDACTICA 1 Construcción de Algoritmos

Identificar problemas y diseñar algoritmos de solución a través del uso de herramientas gráficas (Diagramas de Flujo) o esquema estructurados (pseudocódigos) empleando las técnicas y componentes en una solución computacional, incentivando al empleo de la lógica como el pensamiento más importante.

Competencias específicas (Logro de aprendizaje):

- CEC01: Comprende que es la Gestión de la Configuración del Software y Mantenimiento
- CEC02: Coordina y organiza los grupos de proyectos de software
- CEC03: Propone proyecto para el mantenimiento y desarrollo de Software
- CEC04: Comprende los fundamentos de la Gestión de la Configuración del Software y Mantenimiento
- CEC05: Coordina con su equipo sobre el avance de su proyectos de software en el repositorio

Comprende y analiza los principios de la lógica digital y la representación de la información.

					Evaluación	
Sem	Comp	Contenido	Estrategias didácticas	Actividades	Evidencia	Producto
1ª	CEC01 CEC04	Comprende los fundamentos para crear un algoritmo, empleando pseudocódigo e implementar un programa, muestra orden y disciplina al comunicar los resultados que arroja el algoritmo. Introducción a lenguaje de programación. Proceso de Datos, etapas (entrada, proceso y salida). Estructura de un programa. Definición Algoritmos. Ejemplos. Estructura y funcionamiento General de un Computador	Socializar el sílabo Acuerdos de Convivencia Exposición de ejemplos. Uso de diapositivas Trabajo colaborativo en la solución de casos, Análisis de la solución	Síncrono Establecemos acuerdos de convivencia. Dialogamos sobre el contenido del sílabo. Resuelve problema planteado en clase. Entendimiento de la estructura básica de un algoritmo, aplicando secuencia lógica Asíncrono Prácticas ejemplos de creación de pseudocódigo con herramienta	Analiza y Resuelve problemas simples a través de algoritmos reflejados en diagramas de flujo o en pseudocódigos	Caso práctico incluya diseño de la lógica algorítmica desarrollado en aula basado en problemas
2ª	CEC01 CEC04 CEC05	Comprende los fundamentos para crear un algoritmo, empleando pseudocódigo o flujograma e implementar una aplicación a través de la construcción de un programa donde se use Tipos de Variables, operadores y operadores condicionales	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo Establecer Casos donde se deba aplicar operadores condicionales y analizar sus posibles soluciones	Síncrono Revisa material de clase. Discute ejemplos. Resuelve problemas planteados en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa Asíncrono Práctica creación pseudocódigo problemas básicos	Analiza y diseña soluciones mostrando el uso de variables y operadores condicionales a través de la construcción de una aplicación en un lenguaje de programación	Desarrolla un programa basado en un algoritmo y aplicando variables y operadores condicionales de control Determinar el proyecto grupal, formación de equipos
3ª	CEC01 CEC02 CEC03 CEC04	Comprende los fundamentos para crear un algoritmo y construye programas de	Recuperación de saberes previos. Exposición del docente Uso de diapositivas	Síncrono Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de	Construye algoritmos y genera programas utilizando	Desarrolla un programa aplicando saberes previos y el uso de

		aplicación mostrando el uso de Estructuras de Control Iterativo (Hacer, Mientras)	Trabajo colaborativo Mostrar casos ejemplo en el uso de estructuras Iterativas	manera personal y colaborativa. Propone casos para su solución por los alumnos Asíncrono Practica creación pseudocódigo y código problemas usando iteraciones	adecuadamente estructuras iterativas	estructuras iterativas
4ª	CEC01 CEC05	Comprende y Utiliza adecuadamente los ANIDAMIENTOS en las operaciones condicionales y las estructuras de control Iterativa basadas en una solución algorítmica	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo Mostrar casos ejemplo en el uso de anidamiento de condicionales y de estructuras Iterativas	Síncrono Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal y colaborativa. Propone casos para el uso de estructuras iterativas anidadas Asíncrono Practica creación pseudocódigo y código problemas usando iteraciones anidadas	Construye algoritmos y genera programas utilizando adecuadamente estructuras iterativas con anidamiento	Desarrolla un trabajo práctico aplicando los saberes previos, incluyendo estructuras anidadas Presenta primer avance del proyecto en equipo

UNIDAD DIDÁCTICA 2: Programación Modular y Estructura de Datos Estáticos

Competencias específicas:

Conoce y utiliza adecuadamente los punteros para la manipulación directa en memoria, construye algoritmos y programas usando técnica modular (funciones) en su construcción para mejorar su mantenimiento. Uso de arreglos para agilizar algunos procesos de los programas

- CEC01: Comprende que es la Gestión de la Configuración del Software y Mantenimiento
- CEC02: Coordina y organiza los grupos de proyectos de software
- CEC03: Propone proyecto para el mantenimiento y desarrollo de Software
- CEC04: Comprende los fundamentos de la Gestión de la Configuración del Software y Mantenimiento
- CEC05: Coordina con su equipo sobre el avance de sus proyectos de software en el repositorio

					Evaluación	
Sem	Comp	Contenido	Estrategias didácticas	Actividades	Evidencia	Producto
5ª	CEC01 CEC05	Identifica, define y utiliza las funcionalidades de los punteros, para la manipulación de memoria interna	Exposición de ejemplos. Uso de diapositivas Trabajo colaborativo en la solución de casos con punteros	Síncrono Elaboración de aplicaciones utilizando punteros Planteamiento de problemas donde se usa punteros para su solución Asíncrono Practica uso de punteros en C++	Analiza y construye programas aplicando adecuadamente punteros	Desarrolla trabajo práctico sobre la manipulación de punteros
6ª	CEC01 CEC05	Conoce los pasos para la elaboración de un Menu de Opciones, uso de arreglos de una dimensión (vectores), operaciones sobre vectores	Exposición de ejemplos. Uso de diapositivas Trabajo individual y colaborativo en la solución de casos basados en los temas aprendidos	Síncrono Elaboración de Menu de opciones utilizando condicionales múltiples y plantear ejercicios relacionados a las operaciones sobre vectores Asíncrono Practica uso de vectores con C++	Analiza y construye programa con menú de opciones y aplicando vectores	Programa con uso de menú de opciones y manejo de vectores
7ª	CEC01 CEC02 CEC03	Resuelve soluciones algorítmicas complejas a través del uso de arreglos dentro de estructuras	Exposición de ejemplos sobre el uso de matrices y manejo de cadenas de caracteres, incluyendo funciones	Síncrono Manejo adecuado de funciones y manejo de argumentos en una solución algorítmica compleja, incluyendo funciones con diversos tipos de argumentos	Analiza y construye programas utilizando matrices, estructuras iterativas y	Programas aplicando el uso de matrices, funciones y manejo de cadenas

		iterativas. Conoce y utiliza las instrucciones para el manejo de cadenas de caracteres	Uso de diapositivas, Trabajo individual y Colaborativo	Asíncrono Practica uso de cadenas con C++	funciones	Presenta Segundo avance del proyecto en equipo
8ª		EXAMEN PARCIAL			Analiza y construye una solución algorítmica de complejidad básica empleando los fundamentos, técnicas y herramientas aprendidas	Examen Parcial Teórico Practico
9ª	CEC01 CEC02	Comprende y Aplica los fundamentos de matrices para crear soluciones algorítmicas	Diapositivas, Presentacion de casos practicos	Síncrono Implementacion de algoritmos aplicando el manejo de matrices en su solución Asíncrono Algoritmos y Practica uso de matrices con C++	Analiza y Utiliza adecuadamente técnicas para la implementación del uso de matrices en una solución computacional	Desarrollo práctico a través del planteamiento de un caso de matrices(individual y en equipo)

UNIDAD DIDÁCTICA 3: Estructura Registro y Tipo de datos creado por el Usuario

Conoce, define y utiliza las estructuras de registro, creación de campos, operaciones sobre estructuras de registro (ingresar, modificar, eliminar) datos dentro de una estructura creada

Competencias específicas:

- CEC01: Comprende que es la Gestión de la Configuración del Software y Mantenimiento
- CEC02: Coordina y organizan los grupos de proyectos de software
- CEC03: Propone proyecto para el mantenimiento y desarrollo de Software
- CEC04: Comprende los fundamentos de la Gestión de la Configuración del Software y Mantenimiento
- CEC05: Coordina con su equipo sobre el avance de su proyectos de software en el repositorio

10ª	CEC01 CEC02	Reconoce, define y aplica estructura de registros, arreglos de estructuras en los programas de aplicación	Diapositivas, Presentacion de casos practicos	Síncrono Implementacion de algoritmos aplicando estructuras o registros , uso de vectores de estructuras, estructuras aninadas Asíncrono Algoritmos y Practica uso de Registros con C++	Analiza y Utiliza adecuadamente técnicas para la implementación del uso de estructuras o registros en una solución computacional	Práctica en laboratorio a través de casos prácticos individual y en equipo en el uso de estructuras de registros
11ª	CEC01 CEC02 CEC05	Reconoce, define y utiliza los apuntadores a registros, operaciones básicas .	Diapositivas, Presentacion de casos practicos Trabajo individual y colaborativo	Síncrono Define y utiliza los apuntadores para operar registros Asíncrono Algoritmos y Practica uso de Punteros con Registros en C++	Identifica y define el manejo de apuntadores para realizar operaciones básicas de los registros en un programa	Desarrollo práctico a través del planteamiento de un caso de registros(individual y en equipo) Presenta Tercer avance del proyecto en equipo

UNIDAD DIDÁCTICA 4: Creación y Manejo de Archivos

Conoce, define y genera programas utilizando las instrucciones correspondientes a operaciones básicas sobre los archivos (Altas, Bajas, Actualizaciones y Eliminaciones de Registros)

Competencias específicas:

- CEC01: Comprende que es la Gestión de la Configuración del Software y Mantenimiento
- CEC02: Coordina y organiza los grupos de proyectos de software
- CEC03: Propone proyecto para el mantenimiento y desarrollo de Software
- CEC04: Comprende los fundamentos de la Gestión de la Configuración del Software y Mantenimiento
- CEC05: Coordina con su equipo sobre el avance de su proyectos de software en el repositorio

12 ^a	CEC01 CEC02	Comprende concepto de archivos directos y las operaciones básicas que se realizan sobre ellos.	Diapositivas, Explicación de tipos de archivos, comandos de creación de registros Trabajo individual y colaborativo	Síncrono Reconocimiento de diferentes tipos de archivos, sus estructuras, operaciones sobre ellos Asíncrono Algoritmos y Practica crear de archivos directos con C++	Identifica los diversos comandos para la manipulación de archivos directos	Desarrollo de aplicaciones independientes conteniendo los comandos para operar archivos en forma individual y grupal
13 ^a	CEC01 CEC03	Reconoce y utiliza operaciones de registros sobre archivos en programas de aplicación	Presentación de Diapositivas, Presentación de casos prácticos, Trabajo individual colaborativo	Síncrono Realiza Operaciones de Lectura y grabación de datos desde archivos simples, bajo ciertas condiciones, a través de un programa computacional Asíncrono Algoritmos y Practica operaciones con archivos directos con C++	Genera programa que opere archivos a través de grabación, lectura, modificación y eliminación de campos y registros	Construcción de un programa que maneje las operaciones básicas sobre archivos en forma individual y grupal
14 ^a	CEC03 CEC04	Elabora programas de complejidad intermedia, aplicando los saberes previos	Revisión de los temas contenidos en el curso. Planteamiento de un Caso práctico completo Análisis de su solución en forma colaborativa	Síncrono Plantean diversos tipos de ejercicios de programación, desde simples hasta los que contienen cierta complejidad Seguimiento al desarrollo de las aplicaciones en forma individual y colaborativa Asíncrono Práctica donde se integran los temas con registros y archivos usando C++	Genera programas con niveles de complejidad. Aplica adecuadamente los comandos según el problema asignado	Construye programas de cierta complejidad, aplicando los saberes previos
15 ^a	CEC02 CEC03 CEC04 CEC05	Desarrolla un Proyecto compuesto de un sistema que refleje lo aprendido en el curso	Presentación Expositiva del proyecto desarrollado	Síncrono Analiza, Diseña y Construye una aplicación en equipo basado en un sistema (conjunto de programas) minimamente con los temas contenidos en el curso Asíncrono Desarrollo del proyecto, elaboración del video de presentación	Genera un proyecto basado en el desarrollo de un sistema simple	Evaluación de la Presentación, Contenido, Profundidad, Tiempo del proyecto en equipo
16 ^a		EXAMEN FINAL			Analiza y construye una solución algorítmica de complejidad media empleando los fundamentos, técnicas y herramientas aprendidas	Examen Final Teórico Práctico

5. ESTRATEGIA DIDÁCTICA

El docente desarrollará la asignatura siguiendo los criterios deductivos, inductivos, flexible con la participación activa del estudiante, en este sentido, se utilizará las técnicas de exposición participativa

y desarrollo de solución de problemas en laboratorio.

El estudiante participará activamente a través de intervenciones en las sesiones de teoría y mediante desarrollo de soluciones a problemas con el uso de computadora en las sesiones de laboratorio, individualmente y en equipos.

6. RECURSOS Y MATERIALES

Recursos:

- Aula virtual Google Classroom
- Video conferencia Google Meet
- Repositorio Github
- Lenguaje de Programacion C++
- Herramienta PSeInt, Repl.it

Materiales:

- Diapositivas, casos practicos, prácticas dirigidas y guía de laboratorios

7. EVALUACION DEL APRENDIZAJE

Evaluación Académica	Peso
Examen Parcial (N1)	20%
Práctica Calificada 1 + Práctica Calificada 2 + Práctica Calificada 3 + Presentaciones Informes parciales del Trabajo del Curso e Informe final y Exposición (N2)	60%
Examen Final (N3)	20%

Matriz de evaluación por competencias de la asignatura

Unidades de aprendizaje	Criterios y logros de aprendizaje	Procedimientos (Productos)	Instrumentos de evaluación	Pesos en porcentaje	
				Sesiones	Notas SUM
1	Genera algoritmos básicos aplicando lógica adecuada usando pseudocódigo y/o diagrama de flujo Y construye programas de aplicación simples	Pseudocódigo con PSeINT Uso del C++	Rúbrica	5%	N1 35%
	Construye programa de aplicación con menú de opciones, punteros, funciones y vectores	Uso del C++	Rubrica	5%	
2	Desarrolla un examen Parcial	Examen parcial	Evaluación teórico /práctico	25%	
3	Genera algoritmos y programas aplicando matrices, estructura de registro	Uso del C++	Evaluación escrita	5%	N2 35%
	Genera programas de aplicación en un lenguaje de programación basados en el uso de archivos	Uso del C++	Rubrica	5%	
	Desarrolla un examen Final	Examen Final	Evaluación teórico /práctico	25%	

4	Intervenciones adecuadas, aportes, cumplimiento de tareas, puntualidad comportamiento	Responsabilidad y Comportamiento	Rubrica	5%	N3 30%
	Presenta un proyecto en equipo, reflejando el aprendizaje, técnicas y manejo del lenguaje de programación	Exposición del proyecto	Presentación grupal del proyecto	25%	
Total:				100%	100%
EP: Examen Parcial (20%) ET: Examen Final(20%) EC: Evaluación Continua (40%) + Proyecto (20%) Promedio final = (EPx20%+ EFx20% + ECx60%)					

7.1. RÚBRICAS DE EVALUACIÓN DEL PROYECTO

Criterios	Propositivo	Autónomo	Suficiente	Deficiente	No aceptable
Organización Crea un esquema de caja identificando datos de entrada y salida del problema	Describe claramente lo que va realizar identificando los datos	La descripción no es muy clara, pero se entiende el diseño del programa	La organización esta incompleto y es difícil entender el diseño del programa	Tiene organización deficiente	No tiene organización
	4.0	3.5	3.0	1.0	0.0
Resolución	Utiliza las instrucciones y algoritmos adecuados para resolver el ejercicio	Utiliza las instrucciones y algoritmos necesarios para resolver el ejercicio, aunque no son los adecuados	Utiliza instrucciones y algoritmos que no son acordes para resolver el ejercicio.	Utiliza instrucciones y algoritmos que no resuelven el ejercicio.	No hay resolución
	4.0	3.5	2.0	1.0	0.0
Funcionamiento	El funcionamiento del ejercicio es completo	El funcionamiento del ejercicio no es completo, aunque tiene fallos sin importancia	El funcionamiento del ejercicio tiene fallas importantes	El ejercicio no funciona	No presenta
	5.0	4.0	3.0	2.0	0.0
Identificación de constantes, variables y componentes	Nombra correctamente todos los componentes constantes, variables	Nombra correctamente la mayoría de las constantes, variables y componentes	Nombra correctamente solo algunos componentes	No nombra correctamente ninguna de los componentes.	No presenta
	3.0	2.5	2.0	1.0	0.0
Documentación interna (Buenas practicas)	Aporta una documentación al código para estructurar y entender claramente.	La documentación aportada es apenas para estructurar y/o entender el código	Aporta documentación pero es insuficiente para estructurar y/o entender el código	No aporta documentación ni buenas practicas	No presenta
	2.0	1.5	1.0	0.5	0.0
Interface usuario o salida de datos esperados	La interface es ordenado, claro y tiene todos los componentes y además amigable	La interface tiene todos los componentes y es claro pero no ordenado	La interface tiene todos los componentes pero no es claro, ordenado ni amigable estructurado	A la interface le falta algunos componentes necesarios o utiliza componentes que no son correctos	No presenta
	2.0	1.0	1.0	0.5	0.0

8. REFERENCIAS BIBLIOGRÁFICAS

8.1. BÁSICA

- JOYANES AGUILAR, Luis. (2008) FUNDAMENTOS DE PROGRAMACIÓN. Algoritmos, Estructura de Datos y Objetos. 4ta Ed. McGraw-Hill, Madrid, España.
- Cairo Batistutti, Osvaldo (2008). Fundamentos de la programación. McGraw-Hill. México DF
- JOYANES AGUILAR, Luis y ZAHONERO MARTINEZ, Ignacio. (2005) PROGRAMACION EN C. Metodología, algoritmos y estructura de datos McGraw-Hill. Madrid, España.
- Jensen, Ted (2000) Tutorial sobre Apuntadores y Arreglos en C. es dominio público:
<https://www.cimat.mx/~alram/cpa/pointersC.pdf>
- Gottfried S. Byron (2005) Programacion en C. McGraw-Hill 2daEd. Madrid, Espan
http://profesores.fi-b.unam.mx/ing_gpemn/cpi/fundamentos/pdfs/LenguajeC16.pdf
- SZNAJDLEDER. Pablo A. ALGORITMOS A FONDO Con implementaciones en C y Java. Alfaomega, México D. F.. 2013.

8.2. COMPLEMENTARIA

- SCHOOL, P. C. PEYRIN J. P.. ESQUEMAS ALGORITMICOS FUNDAMENTALES. Secuencia e Iteración 2da Ed., Masson. 2013.
 - OSVALDO CAIRO. METODOLOGIA DE LA PROGRAMACION (Tomo I y II)
 - COMO PROGRAMAR EN C/C++, Deitel & Deitel, Prentice Hall,
 - C/C++ CURSO DE PROGRAMACIÓN, Fco. Javier Ceballos, Alpha Omega
 - PROGRAMACIÓN ESTRUCTURADA EN C , J. R. García-Bermejo, Pearson/Prentice-Hall
<http://www.eduteka.org/pdfdir/AlgoritmosProgramacion.pdf>
-