

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA



VICERRECTORADO ACADÉMICO

FACULTAD DE INGENIERIA DE PRODUCCION Y SERVICIOS
DEPARTAMENTO ACADÉMICO DE INGENIERIA DE SISTEMAS E INFORMATICA

SÍLABO 2023 - A

ASIGNATURA: CIENCIAS DE LA COMPUTACION II

1. INFORMACIÓN ACADÉMICA

Periodo académico:	2023 - A	
Escuela Profesional:	CIENCIA DE LA COMPUTACIÓN	
Código de la asignatura:	1702118	
Nombre de la asignatura:	CIENCIAS DE LA COMPUTACION II	
Semestre:	III (tercero)	
Duración:	17 semanas	
Número de horas (Semestral)	Teóricas:	2.00
	Prácticas:	2.00
	Seminarios:	0.00
	Laboratorio:	4.00
	Teórico-prácticas:	0.00
Número de créditos:	5	
Prerrequisitos:	CIENCIA DE LA COMPUTACION I (1701209)	

2. INFORMACIÓN DEL DOCENTE, INSTRUCTOR, COORDINADOR

DOCENTE	GRADO ACADÉMICO	DPTO. ACADÉMICO	HORAS	HORARIO
MAMANI ALIAGA, ALVARO HENRY	DR.	INGENIERIA DE SISTEMAS E INFORMATICA	0	Lun: 08:50-10:30 Mié: 08:50-10:30
MAMANI ALIAGA, ALVARO HENRY	DR.	INGENIERIA DE SISTEMAS E INFORMATICA	0	Lun: 12:20-14:00 Mié: 12:20-14:00
MAMANI ALIAGA, ALVARO HENRY	DR.	INGENIERIA DE SISTEMAS E INFORMATICA	0	Lun: 10:40-12:20

3. INFORMACIÓN ESPECIFICA DEL CURSO (FUNDAMENTACIÓN, JUSTIFICACIÓN)

Entender el contexto de un programa, desde la computación hasta la ejecución, es de suma importancia para comprender como funcionan los procesos dentro una computadora, esto es más importante hoy en día en el que los ordenadores tienen mas de una unidad central de procesamiento. El curso de Ciencia de

la Computación II pertenece a la línea de programación, donde el alumno debe reforzar lo aprendido en el curso de Ciencia de la Computación I, partiendo del paradigma orientado a objetos y usando como herramienta el lenguaje de programación C++.

4. COMPETENCIAS/OBJETIVOS DE LA ASIGNATURA

Entiende y aplica los conceptos de la programación orientada a objetos.

Conoce, entiende y aplica los conceptos de programación genérica usando templates y traits.

Conoce, entiende y aplica los punteros inteligentes.

Conoce, entiende y aplica la semántica de movimiento.

5. CONTENIDO TEMÁTICO

PRIMERA UNIDAD

Capítulo I: Programación Orientada a Objetos

Tema 01: Punteros, Referencias, Op. New y Delete, Clases

Tema 02: Constructor, Destructor, Constructor copia

Tema 03: Herencia y Polimorfismo

Tema 04: Sobrecarga de operadores: funtores

Capítulo II: Templates

Tema 05: Templates en Funciones

Tema 06: Templates en Clases

Tema 07: Variadic templates

SEGUNDA UNIDAD

Capítulo III: Patrones de Diseño

Tema 08: Patrones creacionales

Tema 09: Patrones estructurales

Tema 10: Patrones de comportamiento

Capítulo IV: Smart Pointers and Memory Management

Tema 11: Introducción a punteros inteligentes

Tema 12: Punteros: shared, unique, weak

Tema 13: Administración de memoria

Tema 14: Custom allocators

Capítulo V: Traits y type deduction

Tema 15: Type traits

Tema 16: type deduction

Capítulo VI: Semántica del Movimiento

Tema 17: Move semantics

Tema 18: Perfect forwarding

TERCERA UNIDAD

Capítulo VII: Metaprogramming

Tema 19: Metaprogramming

Capítulo VIII: Multithreading

Tema 20: Introducción a multithreading

Tema 21: Sincronización y exclusión mutua

Tema 22: Locks y mutexes

Tema 23: Variables de condición

6. PROGRAMACIÓN DE ACTIVIDADES DE INVESTIG. FORMATIVA Y RESPONSABILIDAD SOCIAL

6.1. Métodos

Método expositivo en las clases teóricas.

Método de elaboración conjunta en los seminarios taller y en la elaboración del trabajos de programación/investigación.

6.2. Medios

Pizarra, material de laboratorio, videos, software.

Herramientas tecnológicas para educación: Clases magistrales, LMS (MOODLE), repositorios en línea.

6.3. Formas de organización

Clases Teóricas: Desarrollo de los conceptos teóricos.

Seminarios: Trabajos de investigación y análisis.

Prácticas: Desarrollo de prácticas de programación.

Laboratorio: Aplicación de los conceptos vistos es clases teóricas.

6.4. Programación de actividades de investigación formativa y responsabilidad social

Investigación Formativa: Los alumnos deberán realizar la implementación de una solución real usando lo aprendido en el curso.

Responsabilidad Social: Los alumnos deberán realizar la implementación de una solución real usando lo aprendido en el curso.

7. CRONOGRAMA ACADÉMICO

SEMANA	TEMA	DOCENTE	%	ACUM.
1	Punteros, Referencias, Op. New y Delete, Clases	A. Mamani	8	8.00
2	Constructor, Destructor, Constructor copia	A. Mamani	8	16.00
3	Herencia y Polimorfismo	A. Mamani	8	24.00
4	Sobrecarga de operadores: funtores	A. Mamani	9	33.00
5	Templates en Funciones	A. Mamani		
5	Templates en Clases	A. Mamani		
5	Variadic templates	A. Mamani		

7	Patrones creacionales	A. Mamani	3	36.00
7	Patrones estructurales	A. Mamani	3	39.00
7	Patrones de comportamiento	A. Mamani	3	42.00
8	Introducción a punteros inteligentes	A. Mamani	3	45.00
8	Punteros: shared, unique, weak	A. Mamani	3	48.00
9	Administración de memoria	A. Mamani	3	51.00
9	Custom allocators	A. Mamani	3	54.00
9	Type traits	A. Mamani	3	57.00
9	type deduction	A. Mamani	3	60.00
10	Move semantics	A. Mamani	3	63.00
10	Perfect forwarding	A. Mamani	3	66.00
12	Metaprogramming	A. Mamani	6	72.00
13	Introducción a multithreading	A. Mamani	7	79.00
13	Sincronización y exclusión mutua	A. Mamani	7	86.00
14	Locks y mutexes	A. Mamani	7	93.00
15	Variables de condición	A. Mamani	7	100.00

8. ESTRATEGIAS DE EVALUACIÓN

8.1. Evaluación del aprendizaje

Cada clase teórica será evaluada usando una rúbrica.

Cada practica será evaluada usando una rúbrica.

Cada laboratorio será evaluado usando una rúbrica.

Todas las notas obtenidas en los puntos anteriores se promediarán para obtener la nota de Evaluación Continua (EC)

8.2. Cronograma de evaluación

EVALUACIÓN	FECHA DE EVALUACIÓN	EXAMEN TEORÍA	EVAL. CONTINUA	TOTAL (%)
Primera Evaluación Parcial	15-05-2023	12%	18%	30%
Segunda Evaluación Parcial	26-06-2023	12%	18%	30%
Tercera Evaluación Parcial	31-07-2023	16%	24%	40%
TOTAL				100%

9. REQUISITOS DE APROBACIÓN DE LA ASIGNATURA

El alumno tendrá derecho a observar o en su defecto a ratificar las notas consignadas en sus evaluaciones, después de ser entregadas las mismas por parte del profesor, salvo el vencimiento de plazos para culminación del semestre académico, luego del mismo, no se admitirán reclamaciones, alumno que no se haga presente en el día establecido, perderá su derecho a reclamo.

Para aprobar el curso el alumno debe obtener una nota igual o superior a 11, en el promedio final. El redondeo, solo se efectuara en el cálculo del promedio final, quedado expreso, que las notas parciales, no se redondearan individualmente.

El alumno que no tenga alguna de sus evaluaciones y no haya solicitado evaluación de rezagados en el plazo oportuno, se le considerará como abandono.

El alumno quedará en situación de "abandono" si no presenta Trabajo Final.

El alumno quedara en situación de "abandono" si el porcentaje de asistencia es menor al ochenta (80%) por ciento en las actividades que requieran evaluación continua (Practicass, laboratorio, etc).

El Promedio Final (PF) se calcula con la siguiente fórmula: $PF = EC1*0.18 + EP1*0.12 + EC2*0.18 + EP2*0.12 + EC3*0.24 + EP3*0.16$.

10. BIBLIOGRAFIA: AUTOR, TÍTULO, AÑO, EDITORIAL

10.1. Bibliografía básica obligatoria

Deitel, P. and Deitel, H. (2017). C++ How to Program. Pearson, 10th edition.

The C++ Notes for Professionals book, 2018

10.2. Bibliografía de consulta

Bjarne Stroustrup, "The C++ Programming Language" (4th edition), Addison-Wesley Professional, 2013.

Scott Meyers, "Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14", O'Reilly Media, 2014.

Herb Sutter, "C++ Coding Standards: 101 Rules, Guidelines, and Best Practices", Addison-Wesley Professional, 2004.

Arequipa, 29 de Mayo del 2023

MAMANI ALIAGA, ALVARO HENRY