

SILABO

FACULTAD
ESCUELA PROFESIONAL
PROGRAMA DE ESTUDIOS

INGENIERIA ESTADISTICA E INFORMATICA
INGENIERIA ESTADISTICA E INFORMATICA
CARRERA PURA

I. INFORMACIÓN GENERAL

1.1 Identificación Académica

a)	Curso	LENGUAJES DE PROGRAMACION II
b)	Código	EST305
c)	Prerequisito	EST303 - LENGUAJES DE PROGRAMACION I
d)	Número de Horas	03h teóricas, 02h prácticas, Total 05 horas
e)	Créditos	04
f)	Número de Horas virtuales	04
g)	Año y Semestre Académico	2025-II
h)	Ciclo de Estudios	IV
i)	Duración	Del 25 de Agosto al 19 de Diciembre del 2025 (17 semanas)
j)	Área Curricular	Estudios de especialidad
k)	Características del Curso	I+D+I, experiencia pre profesional

1.2 Docente

a)	Apellidos y Nombres	COYLA IDME LEONEL
b)	Condición y Categoría	Asociado a T.C.
c)	Especialidad	Estadística e Informática
a)	Apellidos y Nombres	MAMANI CALISAYA MILTON VLADIMIR
b)	Condición y Categoría	Contratado B-2
c)	Especialidad	Estadística - Informática

1.3 Ambiente donde se realizó el aprendizaje

- a) a) E.I. – A. 102, Pab. B

II. SUMILLA

El curso de Lenguajes de Programación II, correspondiente al área de estudios de especialidad, es de carácter teórico-práctico y está orientado a profundizar en paradigmas de programación avanzada como la programación orientada a objetos (POO) y la programación modular, aplicando estos conocimientos al desarrollo de soluciones informáticas escalables y eficientes. Los estudiantes desarrollarán habilidades en lenguajes para construir aplicaciones orientadas a objetos. Este curso se enfoca en la arquitectura de software que facilita la reutilización, modularidad, y escalabilidad, alineándose con los estándares internacionales de calidad del software. El curso está dividido en dos unidades:

- Fundamentos de Programación Orientada a Objetos y Modularidad
- Principios avanzados de POO

III. PERFIL DEL EGRESADO EN RELACIÓN AL CURSO

CE1. Diseña y desarrolla soluciones informáticas, aplicaciones web y soluciones empresariales bajo estándares internacionales para aportar con el desarrollo tecnológico, asumiendo una actitud científica y responsable utilizando principios, técnicas, herramientas y métodos que aseguren la funcionalidad y calidad del software.

IV. LOGRO DE APRENDIZAJE DEL CURSO

Implementa programas orientados a objetos aplicando principios como encapsulamiento, herencia y polimorfismo, utilizando un lenguaje de programación orientado a objetos y buenas prácticas de codificación.

V. TRATAMIENTO DE UNIDADES DIDÁCTICAS

UNIDAD 1		Fundamentos de la Programación Orientada a Objetos
LOGROS DE APRENDIZAJE DE LA UNIDAD Comprende los conceptos fundamentales de la programación orientada a objetos, como clases, objetos y métodos, a partir del análisis de problemas simples y el uso de ejemplos prácticos en un entorno de desarrollo		
TIEMPO DE DESARROLLO		Del 25 de Agosto al 27 de Octubre del 2025 (Total 45 horas)
HORAS DE ENSEÑANZA VIRTUAL/UNIDAD		02
SEMANAS	CRITERIOS DE DESEMPEÑO	CONOCIMIENTOS
Semana 1	Explica los conceptos básicos de programación y las diferencias entre los paradigmas estructurado y orientado a objetos mediante ejemplos en un lenguaje de programación.	Introducción a POO. Conceptos básicos de programación, diferencias entre programación estructurada y orientada a objetos
Semana 2	Identifica los elementos fundamentales del paradigma orientado a objetos — como clases, objetos, atributos y métodos— mediante el análisis y desarrollo de ejemplos básicos en el lenguaje de programación elegido.	Paradigma de Objetos. Objetos, clases, atributos, métodos

Semana 3	Escribe clases y crea instancias de objetos utilizando la sintaxis básica del lenguaje de programación seleccionado.	Sintaxis básica en el lenguaje elegido. Creación de clases, instanciación de objetos
Semana 4	Encapsulamiento, modificadores de acceso, getters y setters aplican para proteger y controlar el acceso a los atributos de una clase utilizando la sintaxis del lenguaje de programación seleccionado.	Encapsulamiento. Modificadores de acceso, getters y setters
Semana 5	El estudiante implementa métodos con parámetros, valores de retorno y sobrecarga, aplicando correctamente la sintaxis del lenguaje de programación seleccionado.	Métodos y sobrecarga. Parámetros, retorno, sobrecarga de métodos
Semana 6	Constructores, destructores e inicialización de objetos implementan en el diseño de clases, considerando el ciclo de vida del objeto y utilizando la sintaxis correcta del lenguaje de programación seleccionado.	Constructores y destructores. Inicialización y ciclo de vida de un objeto
Semana 7	Relaciones entre clases (asociación, agregación y composición) modelan utilizando diagramas y código, según la sintaxis del lenguaje de programación orientado a objetos seleccionado.	Relaciones entre clases. Asociación, agregación, composición
Semana 8	Proyecto intermedio desarrollan integrando los conceptos y técnicas aprendidas en la Unidad I, mediante la construcción de una aplicación funcional en el lenguaje de programación orientado a objetos.	Ejercicio práctico de unidad (mini sistema con varias clases)
Semana 9	Proyecto intermedio desarrollan integrando los conceptos y técnicas aprendidas en la Unidad I, mediante la construcción de una aplicación funcional en el lenguaje de programación orientado a objetos.	Proyecto intermedio: Aplicaciones de todos los conceptos de la Unidad I
PORCENTAJE DE AVANCE ACADÉMICO DE LA UNIDAD: 100%		

* El docente deberá programar “actividades de retroalimentación” los cuales se desarrollarán a lo largo de cada unidad didáctica con el fin de asegurar los adecuados aprendizajes de los estudiantes.

UNIDAD 2		Avances y aplicaciones de la POO
LOGROS DE APRENDIZAJE DE LA UNIDAD Analiza los avances y aplicaciones de la programación orientada a objetos en el desarrollo de software moderno, considerando paradigmas actuales, buenas prácticas y contextos tecnológicos diversos.		
TIEMPO DE DESARROLLO		Del 27 de Octubre al 19 de Diciembre del 2025 (Total 40 horas)
HORAS DE ENSEÑANZA VIRTUAL/UNIDAD		02
SEMANAS	CRITERIOS DE DESEMPEÑO	CONOCIMIENTOS
Semana 10	El estudiante explica el concepto de herencia en POO, utilizando ejemplos prácticos que demuestren su aplicación en el desarrollo de software.	Herencia en POO. Concepto de herencia, Herencia simple y múltiple, Uso de super() en Python, Ejercicios sobre jerarquías de clases
Semana 11	El estudiante explica el concepto de polimorfismo utilizando ejemplos que demuestren su aplicación en la Programación Orientada a Objetos.	Polimorfismo. Definición de polimorfismo, Sobrecarga y sobrescritura de métodos, Aplicaciones del polimorfismo en el desarrollo de software
Semana 12	El estudiante explica el concepto de clases abstractas e interfaces, diferenciando sus usos y aplicaciones en la Programación Orientada a Objetos.	Clases Abstractas e Interfaces. Diferencia entre clases abstractas e interfaces. Implementación en Python. Casos de uso de interfaces y clases abstractas
Semana 13	El estudiante maneja las excepciones en POO, implementando correctamente estructuras de control para el tratamiento de errores en el código.	Manejo de Excepciones en POO. Introducción a las excepciones. Uso de try, except (Python). Creación de excepciones personalizadas
Semana 14	El estudiante implementa la programación genérica y utiliza colecciones, aplicando tipos parametrizados y estructuras de datos eficientes en el desarrollo de software.	Programación Genérica y Colecciones. Introducción a la programación genérica. Uso de listas, diccionarios y conjuntos en POO. Clases genéricas en Java y estructuras de datos en POO
Semana 15	El estudiante aplica los principios SOLID en POO, justificando su importancia en el diseño de software modular y mantenible.	Principios SOLID en POO. Principios de diseño de software. Aplicación de SOLID en la programación orientada a objetos
Semana 16	El estudiante desarrolla un proyecto final aplicando los principios de la Programación Orientada a Objetos para resolver un problema específico.	Proyecto Final. Desarrollo de una aplicación completa utilizando POO. Aplicación de herencia, polimorfismo y encapsulamiento
Semana 17	El estudiante realiza la evaluación final demostrando el dominio de los conceptos y habilidades adquiridas durante el curso.	Evaluación Final. Presentación del proyecto final. Examen teórico sobre temas avanzados de POO.
PORCENTAJE DE AVANCE ACADÉMICO DE LA UNIDAD: 100%		

* El docente deberá programar “actividades de retroalimentación” los cuales se desarrollarán a lo largo de cada unidad didáctica con el

fin de asegurar los adecuados aprendizajes de los estudiantes.

VI. ESTRATEGIAS METODOLÓGICAS

6.1 De Enseñanza

- Aprendizaje basado en problemas
- Video conferencias

6.2 De Aprendizaje

- Prácticas de laboratorio
- Talleres
- Debate

6.3 De Investigación Formativa

- Actividades de investigación bibliográfica en páginas web.

6.4 De Responsabilidad Social Universitaria

- Compartir propuestas de reflexión y acción estratégica para el desarrollo humano sostenible en las redes sociales.

6.5 De Enseñanza Virtual

- Utilización del aula virtual y el uso de recurso de tecnología e información
- Uso de Google meet - videoconferencia
- Uso de Google drive, Whatsapp, Facebook
- Uso de la IA

VII. MEDIOS Y MATERIALES DIDÁCTICOS

- Medios: Auditivo, visual y audiovisual
- Materiales: Laptop, smartphone, navegadores, editores de texto, diapositivas, videos

VIII. EVALUACIÓN DEL APRENDIZAJE

8.1 Logro de aprendizaje, evidencias de desempeño, ponderación, técnicas e instrumentos de evaluación.

UNIDAD	LOGROS DE APRENDIZAJE	EVIDENCIAS DESEMPEÑO: De acción, objeto o producto (%)	PONDERACIÓN (Obligatorio en base a 100%)	TÉCNICAS	INSTRUMENTOS
1	Comprende los conceptos fundamentales de la programación orientada a objetos, como clases, objetos y métodos, a partir del análisis de problemas simples y el uso de ejemplos prácticos en un entorno de desarrollo	Evidencia 1: Práctica calificada de programación modular (40%) Evidencia 2: Presentación de proyectos de innovación (40%) Evidencia 3: Informe de trabajos (20%)	50%	Ejercicio práctico (mini sistema con clases)	Rúbrica de evaluación (para validar uso de clases, objetos, métodos y sintaxis)
2	Analiza los avances y aplicaciones de la programación orientada a objetos en el desarrollo de software moderno, considerando paradigmas actuales, buenas prácticas y contextos tecnológicos diversos.	Evidencia 4: Práctica calificada de Python (40%) Evidencia 5: Presentación de proyectos de innovación (40%) Evidencia 6: Informe de trabajos (20%)	50%	Proyecto integrador (aplicación funcional con herencia, polimorfismo, relaciones entre clases)	Rúbrica de proyecto (criterios: aplicación de principios POO, estructura, funcionalidad)

8.2 Evidencias de aprendizaje del semestre académico.

LOGRO DE APRENDIZAJE DEL CURSO	EVIDENCIAS: ACCIÓN, OBJETO o PRODUCTO		FECHA DE PRESENTACIÓN
Implementa programas orientados a objetos aplicando principios como encapsulamiento, herencia y polimorfismo, utilizando un lenguaje de programación orientado a objetos y buenas prácticas de codificación.	Acción	Implementa clases, objetos y métodos aplicando principios de POO.	La penúltima semana del semestre académico
	Objeto	Codifica estructuras orientadas a objetos con sintaxis correcta.	
	Producto	Desarrolla una aplicación funcional con conceptos de POO.	

- Evidencias que deben ser socializados con la comunidad como parte de la responsabilidad social universitaria. Por la emergencia sanitaria podría optarse por la difusión vía página web y redes sociales.

8.3 Calificación

La fórmula para la obtención del promedio final del curso es la siguiente:

$$\text{Promedio Final} = (50\%)IUPP + (50\%)IIUPP$$

Donde:

IUPP : Primero unidad promedio parcial

IIUPP : Segundo unidad promedio parcial

La fórmula para la obtención del promedio final del curso es la siguiente:

$$IUPP = ((EV1)40\% + (EV2)40\% + (EV3)20\%) / 100\%$$

$$IIUPP = ((EV4)40\% + (EV5)40\% + (EV6)20\%) / 100\%$$

EV1 = Práctica calificada programación modular.; EV2 = Presentación de proyectos de innovación. EV3 = Informe de trabajos;

EV4 = Práctica calificada de Python. EV5 = Presentación de proyectos de innovación; EV6 = Informe de trabajos.

$$PF = (IUPP + IIUPP) / 2$$

IX. FUENTES DE INFORMACIÓN

9.1 Bibliográficas

Básica

Velarde de Barraza Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol Felícita. (2006). Introducción a la programación orientada a objetos. Primera edición. Pearson Educación México. <https://travezurasdeltravieso.wordpress.com/wp-content/uploads/2014/02/introduccion-a-la-programacion-orientada-a-objetos.pdf>.

Andrés Marcelo Salinas Copo, Luis Tarquino Vinueza Rodríguez, Jorge Giovanni Valdiviezo Rodríguez, René Omar Villa López, David Leonardo Guevara Aulestia, Cecilia Estefanía Urquiza Álvarez, Deysi Emilene Moya Ibarra, Irma Victoria Espín Mendoza. (2025). Fundamentos de Programación Orientada a Objetos (Poo). Centro de Investigación y Desarrollo Ecuador.

<https://repositorio.cidecuador.org/bitstream/123456789/3182/3/LIBRO%20DE%20PROGRAMACION%20ORIENTADA%20A%20OBJETOS.pdf>

Complementarias

López Román Leobardo. (2013). Metodología de la programación orientada a objetos. Segunda Edición. Alfaomega. https://irp.cdn-website.com/c4d16642/files/uploaded/Metodologia_de_la_programacion_orientada.pdf

Bermón Angarita Leonardo (2021) Ejercicios de programación orientada a objetos con Java y uml. Editorial Universidad Nacional de Colombia. Primera edición. https://fadmon.unal.edu.co/fileadmin/user_upload/investigacion/centro_editorial/libros/ejercicios%20de%20programacion.pdf

Electrónicas

Esparza Cruz Nelly Karina. (2025) "Programación Orientada a Objetos para principiantes" <<https://libros.utb.edu.ec/index.php/utb/catalog/book/12> > [Consulta: 20 de marzo 2025]

Producción intelectual del docente relacionado con el curso

Coyla Idme Leonel (2021) Métodos Numéricos con Python 105p ISBN 978-612-00-8103-7

Puno, Diciembre del 2025