

SÍLABO

PROGRAMACION ORIENTADA A OBJETOS (100000SI34)

2024 - Ciclo 1 Marzo

1. DATOS GENERALES

1.1. Carrera:	Ingeniería de Sistemas e Informática Ingeniería de Software
1.2. Créditos:	3
1.3. Enseñanza de curso:	Virtual en vivo
1.4. Horas semanales:	4

2. FUNDAMENTACIÓN

En la actualidad, en el mundo del desarrollo de software, el enfoque orientado a objetos se ha convertido en el predominante. Por ello, el curso permitirá que el estudiante posea los conocimientos sólidos de análisis, diseño, construcción e implementación de software basados en computadoras aplicando metodologías y paradigmas de desarrollo orientado a objetos.

3. SUMILLA

El curso es de naturaleza práctica. Presenta conceptos claves del diseño orientado a objetos, análisis y diseño orientado a objetos, utilización de casos de uso, modelado de dominios, creación de aplicaciones con Java, herencia y composición en el diseño orientado a objetos y patrones de diseño orientados a objetos.

4. LOGRO GENERAL DE APRENDIZAJE

Al finalizar el curso, el estudiante construye soluciones informáticas mediante lenguajes de programación orientados a objetos, considerando los requerimientos planteados.

5. UNIDADES Y LOGROS ESPECÍFICOS DE APRENDIZAJE

Unidad de aprendizaje 1: Paradigma de programación orientada a objetos, herencia y relaciones entre clases..	Semana 1,2,3,4 y 5
Logro específico de aprendizaje: Al finalizar la unidad, el estudiante aplica los conceptos básicos de la programación orientada a objetos, el concepto de herencia, relaciones entre clases, en la solución de problemas usando Java.	
Temario: <ul style="list-style-type: none">• Diseño de diagramas de clases UML• Clases y Objetos Atributos y métodos Instanciación.• Encapsulación. Modificadores de acceso.• Constructores Sobre escritura de métodos.• Relaciones de asociación.• Relaciones de agregación y composición.• Jerarquía de clases. Herencia.• Trabajar con subclases y superclases.• Diseño de diagrama de clases del UML usando herencia simple y herencia múltiple• Sustitución de métodos de la superclase	
Unidad de aprendizaje 2: Polimorfismo y principios de diseño program-to-an-interface..	Semana 6,7,8,9,10 y 11
Logro específico de aprendizaje: Al finalizar la unidad, el estudiante aplica el polimorfismo en la solución de problemas e implementa soluciones que utilicen eficientemente el principio de diseño Program-to-an-interface con Java.	

Temario:

- Clases abstractas. Creación y ampliación de clases abstractas
- Polimorfismo. Diseño del diagrama de clases del UML usando polimorfismo.
- Implementación de programas usando polimorfismo.
- Interfaces.
- Principio de diseño Program-to-an-interface.
- Diseño del diagrama de clases del UML usando clases abstractas e interfaces.
- Clases abstractas e interfaces
- Colecciones: LinkedList, ArrayList, Hashmap
- Trabajar con listas de elementos. Procesamiento de colecciones de objetos.

Unidad de aprendizaje 3:

Conceptos avanzados de programación orientada a objetos..

Semana 12,13,14,15,16,17 y 18

Logro específico de aprendizaje:

Al finalizar la unidad, el estudiante soluciona problemas aplicando los conceptos de clases genéricas y colecciones y patrones usando Java.

Temario:

- Clases genéricas.
- Introducción a las expresiones lambda. Uso de expresiones Lambda como argumento de métodos.
- Métodos lambda con objetos.
- Introducción a Patrones GoF.
- Clases Persistentes. Serialización en archivos XML
- Programación con clases persistentes y bases de datos.
- Programación orientado a objetos con bases de datos.
- Patrones de diseño básicos: singleton, adaptador, iterator.
- Patrones de diseño básicos : builder, facade, observer, state.

6. METODOLOGÍA

El curso se desarrolla a través de la plataforma virtual de aprendizaje que se usa como principal medio para el desarrollo de las sesiones sincrónicas que son complementadas con recursos y materiales que se publican a lo largo del curso para fomentar el desarrollo de aprendizajes significativos. Por otro lado, el estudiante dispone en la plataforma virtual de aprendizaje de un espacio de foro de consultas para resolver las dudas académicas a lo largo del curso. Finalmente, las actividades de evaluación se desarrollan de acuerdo con lo señalado en el sílabo a través de la plataforma virtual de aprendizaje (aprendizaje para la era digital).

7. SISTEMA DE EVALUACIÓN

El cálculo del promedio final se hará de la siguiente manera:

$$(10\%)APF1 + (20\%)APF2 + (10\%)TA1 + (20\%)APF3 + (10\%)TA2 + (30\%)PROY$$

Donde:

Tipo	Descripción	Semana	Observación
APF1	AVANCE DE PROYECTO FINAL 1	5	Evaluación flexible. Proyecto Final con un avance al 30%.
APF2	AVANCE DE PROYECTO FINAL 2	10	Evaluación flexible. Proyecto Final con un avance al 50%.
TA1	TAREA ACADÉMICA 1	14	Evaluación flexible. Trabajo investigación: Clases persistentes con Python, C#, Scala, resolución de ejercicios
APF3	AVANCE DE PROYECTO FINAL 3	15	Evaluación flexible. Proyecto Final con un avance al 70%.
TA2	TAREA ACADÉMICA 2	16	Evaluación flexible. Trabajo investigación: Clases persistentes con base de datos en Python, C#, Scala.
PROY	PROYECTO FINAL	18	Evaluación flexible. Proyecto Final con un avance al 100%.

Indicaciones sobre Fórmulas de Evaluación:

1. La nota mínima aprobatoria final es de 12.

2. En este curso, no aplica examen rezagado.
3. En este curso, ninguna nota se reemplaza.
4. En las evaluaciones flexibles, el estudiante debe elegir si desarrollarla de manera individual o grupal.

8. FUENTES DE INFORMACIÓN

Bibliografía Base:

- MORENO PÉREZ, J. *Programación orientada a objetos*. RA-MA Editorial. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=31933>

Bibliografía Complementaria:

- Ricardo Ruiz Rodríguez. *Fundamentos de la programación orientada a objetos: una aplicación a las estructuras de datos en Java*. El Cid Editor. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36792>
- Vélez Serrano, José. *Diseñar y programar, todo es empezar: una introducción a la Programación Orientada a Objetos usando UML y Java*. Dykinson. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=37706>
- Blasco, Francisco. *Programación orientada a objetos en Java*. Ediciones de la U. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=37706>
- Moreno Pérez, Juan. *Programación orientada a objetos*. RA-MA Editorial. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=37705>

9. COMPETENCIAS

Carrera	Competencias específicas
Ingeniería de Sistemas e Informática	<ul style="list-style-type: none"> Soluciones Informáticas
Ingeniería de Software	<ul style="list-style-type: none"> Desarrollo de software Soluciones informáticas

10. CRONOGRAMA DE ACTIVIDADES

Unidad de aprendizaje	Semana	Sesión	Tema	Actividades y evaluaciones
Unidad 1	1	1	Diseño de diagramas de clases UML	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		2	Clases y Objetos Atributos y métodos Instanciación.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	2	3	Encapsulación. Modificadores de acceso.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		4	Constructores Sobre escritura de métodos.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	3	5	Relaciones de asociación.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades

Paradigma de programación orientada a objetos, herencia y relaciones entre clases.		6	Relaciones de agregación y composición.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	4	7	Jerarquía de clases. Herencia.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		8	Trabajar con subclases y superclases.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	5	9	Diseño de diagrama de clases del UML usando herencia simple y herencia múltiple	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		10	Sustitución de métodos de la superclase	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
			Evaluación	<ul style="list-style-type: none"> AVANCE DE PROYECTO FINAL 1
Unidad 2 Polimorfismo y principios de diseño program-to-an-interface.	6	11	Clases abstractas. Creación y ampliación de clases abstractas	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		12	Polimorfismo. Diseño del diagrama de clases del UML usando polimorfismo.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	7	13	Implementación de programas usando polimorfismo.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	8	14	Interfaces.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		15	Principio de diseño Program-to-an-interface.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	9	16	Diseño del diagrama de clases del UML usando clases abstractas e interfaces.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades

	10	17	Clases abstractas e interfaces	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		18	Colecciones: LinkedList, ArrayList, Hashmap	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
			Evaluación	<ul style="list-style-type: none"> AVANCE DE PROYECTO FINAL 2
	11	19	Trabajar con listas de elementos. Procesamiento de colecciones de objetos.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
Unidad 3 Conceptos avanzados de programación orientada a objetos.	12	20	Clases genéricas.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	13	21	Introducción a las expresiones lambda. Uso de expresiones Lambda como argumento de métodos.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		22	Métodos lambda con objetos.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
	14	23	Introducción a Patrones GoF.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
			Evaluación	<ul style="list-style-type: none"> TAREA ACADÉMICA 1
	15	24	Clases Persistentes. Serialización en archivos XML	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
		25	Programación con clases persistentes y bases de datos.	<ul style="list-style-type: none"> Exposición de los temas de clase Desarrollo de actividades
			Evaluación	<ul style="list-style-type: none"> AVANCE DE PROYECTO FINAL 3
			Programación orientado a objetos con bases de datos.	<ul style="list-style-type: none"> Exposición de los temas de clase

	16	26		<ul style="list-style-type: none"> • Desarrollo de actividades
			Evaluación	<ul style="list-style-type: none"> • TAREA ACADÉMICA 2
	17	27	Patrones de diseño básicos: singleton, adaptador, iterator.	<ul style="list-style-type: none"> • Exposición de los temas de clase • Desarrollo de actividades
		28	Patrones de diseño básicos : builder, facade, observer, state.	<ul style="list-style-type: none"> • Exposición de los temas de clase • Desarrollo de actividades
	18	29	Evaluación	<ul style="list-style-type: none"> • PROYECTO FINAL