

SÍLABO

PROGRAMACIÓN ORIENTADA A OBJETOS (100000I55N)

2021 - Ciclo 2 Agosto

1. DATOS GENERALES

- 1.1. Carrera: Ingeniería de Sistemas e Informática
 Ingeniería de Software
 Ingeniería de Redes y Comunicaciones
 Ingeniería de Seguridad y Auditoría Informática
 Ingeniería de Diseño Computacional
- 1.2. Créditos: 3
- 1.3. Modalidad: Presencial (adaptado a la educación no presencial)
- 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 lo tanto es importante para el estudiante poseer 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 teórico-práctico y 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 APRENDIZAJE¹

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"> • Clases y Objetos Atributos y métodos Instanciación. • Constructores Sobre escritura de métodos. • Jerarquía de clases. Herencia. • Diseño de diagramas de clases UML Encapsulación. Modificadores de acceso. • Relaciones de agregación y composición. • Trabajar con subclases y superclases. • Relaciones de asociación. • Sustitución de métodos de la superclase • Diseño de diagrama de clases del UML usando herencia simple y herencia múltiple 	
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: <ul style="list-style-type: none"> • Clases abstractas. Creación y ampliación de clases abstractas • Clases abstractas e interfaces • Trabajar con listas de elementos. Procesamiento de colecciones de objetos. • Diseño del diagrama de clases del UML usando clases abstractas e interfaces. • Implementación de programas usando polimorfismo. • Interfaces. • Polimorfismo. Diseño del diagrama de clases del UML usando polimorfismo. Colecciones: LinkedList, ArrayList, Hashmap Principio de diseño • Program-to-an-interface. 	
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: <ul style="list-style-type: none"> • Clases Persistentes. Serialización en archivos XML • Introducción a las expresiones lambda. Uso de expresiones Lambda como argumento de métodos. • Clases genéricas. • Programación orientado a objetos con bases de datos. • Entrega de Proyecto Final (4) con un avance de 100%. • Programación con clases persistentes y bases de datos. • Patrones de diseño básicos: singleton, adaptador, iterator. • Métodos lambda con objetos. • Patrones de diseño básicos : builder, facade, observer, state. Introducción a Patrones GoF.	

6. METODOLOGÍA

El curso se desarrolla a través de la plataforma Canvas, 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. Para fomentar la aplicación de los contenidos, se desarrollarán ejercicios prácticos a lo largo del curso a través de diferentes herramientas propuestas por el docente. Por otro lado, el estudiante contará con un espacio de foro de consultas para resolver las dudas académicas a lo largo del curso. Finalmente, las actividades de evaluación se desarrollarán de acuerdo a lo señalado en el sílabo a través de la plataforma Canvas.

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

PROY	PROYECTO FINAL	18	Proyecto Final con un avance al 100%.
------	----------------	----	---------------------------------------

Indicaciones sobre Fórmulas de Evaluación:

1. La nota mínima aprobatoria es 12 (doce).

8. FUENTES DE INFORMACIÓN**Bibliografía Base:**

- BARNES. PROG ORIENTADA A OBJS C/JAVA U SANDO BLUEJ

Bibliografía Complementaria:

- BERNAL BERMUDEZ, JESÚS Programación orientada a objetos con Java , <https://www.etsisi.upm.es/sites/default/files/>
- BARNES, DAVID (2013) Programación Orientada a Objetos con Java usando BlueJ, Pearson
- FLÓREZ FERNÁNDEZ, HÉCTOR ARTURO (2012) Programación orientada a objetos usando Java, Macro
- FLORES CUETO, JUAN Y ACUÑA FLORES, CARLOS (2014) Método de las 6'D. Modelamiento – Algoritmo – Programación. Tomo III. Enfoque orientado a los arreglos de objetos, listas enlazadas y archivos de datos, Área de documentación del MIW. Universidad Politécnica de Madrid
- TORRES REMON, MANUEL A. (2013) Desarrollo de aplicaciones con Java, Prentice Hall

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
Ingeniería de Redes y Comunicaciones	<ul style="list-style-type: none"> • Soluciones Informáticas
Ingeniería de Seguridad y Auditoría Informática	
Ingeniería de Diseño Computacional	<ul style="list-style-type: none"> • Proyectos de Ingeniería de Diseño Computacional Diseño • Asistido por Computadora (CAD)

10. CRONOGRAMA DE ACTIVIDADESⁱⁱ

Unidad de aprendizaje	Semana	Sesión	Tema	Actividades y evaluaciones
Unidad 1 Paradigma de	1	1	Diseño de diagramas de clases UML	<ul style="list-style-type: none"> • Prueba de entrada de 15 minutos. • Desarrollar ejercicios diagramas de clases
		2	Clases y Objetos Atributos y métodos Instanciación.	<ul style="list-style-type: none"> • Desarrollar ejercicios usando Clases y Objetos • Desarrollar ejercicios usando atributos y métodos
	2	3	Encapsulación. Modificadores de acceso.	<ul style="list-style-type: none"> • Desarrollar ejercicios con Clases, Objetos y encapsulamiento
		4	Constructores Sobre escritura de métodos.	<ul style="list-style-type: none"> • Ejercicios con constructores

programación orientada a objetos, herencia y relaciones entre clases.

	3	5	Relaciones de asociación.	<ul style="list-style-type: none"> Desarrollar ejercicios con clases y relaciones de asociación
		6	Relaciones de agregación y composición.	<ul style="list-style-type: none"> Desarrollar ejercicios con clases y relaciones de agregación y composición.
	4	7	Jerarquía de clases. Herencia.	<ul style="list-style-type: none"> Desarrollar ejercicios con clases y herencia.
		8	Trabajar con subclases y superclases.	<ul style="list-style-type: none"> Desarrollar ejercicios con subclases y superclases
	5	9	Diseño de diagrama de clases del UML usando herencia simple y herencia múltiple	<ul style="list-style-type: none"> Desarrollar ejercicios con UML para diagrama de clases con Herencia simple
		10	Sustitución de métodos de la superclase	<ul style="list-style-type: none"> Desarrollar ejercicios con sustitución de métodos de la superclase. AVANCE DE PROYECTO FINAL 1
	6	11	Clases abstractas. Creación y ampliación de clases abstractas	<ul style="list-style-type: none"> Desarrollar ejercicios con clases abstractas
		12	Polimorfismo. Diseño del diagrama de clases del UML usando polimorfismo.	<ul style="list-style-type: none"> Desarrollar ejercicios con UML para diagramas de clases con polimorfismo.
	7	13	Implementación de programas usando polimorfismo.	<ul style="list-style-type: none"> Desarrollar ejercicios con polimorfismo
	8	14	Interfaces.	<ul style="list-style-type: none"> Desarrollar ejercicios con Interfaces
		15	Principio de diseño Program-to-aninterface.	<ul style="list-style-type: none"> Desarrollar ejercicios con Interfaces
Unidad 2 Polimorfismo y principios de diseño program-to-aninterface.	9	16	Diseño del diagrama de clases del UML usando clases abstractas e interfaces.	<ul style="list-style-type: none"> Desarrollar ejercicios con UML para diagramas de clases con Interfaces y clases abstractas

	10	17	Clases abstractas e interfaces	<ul style="list-style-type: none"> Desarrollar ejercicios usando clases abstractas e interfaces
		18	Colecciones: LinkedList, ArrayList, Hashmap	<ul style="list-style-type: none"> Desarrollar ejercicios usando ArrayList, LinkedList, Hashmap AVANCE DE PROYECTO FINAL 2
		11	19	Trabajar con listas de elementos. Procesamiento de colecciones de objetos. <ul style="list-style-type: none"> Desarrollar ejercicios realizando operaciones básicas en colecciones de elementos.
Unidad 3 Conceptos avanzados de programación orientada a objetos.	12	20	Clases genéricas.	<ul style="list-style-type: none"> Desarrollar ejercicios usando clases genéricas.
	13	21	Introducción a las expresiones lambda. Uso de expresiones Lambda como argumento de métodos.	<ul style="list-style-type: none"> Desarrollar ejercicios usando expresiones lambda.
		22	Métodos lambda con objetos.	<ul style="list-style-type: none"> Desarrollar ejercicios usando expresiones lambda con colecciones de objetos.
	14	23	Introducción a Patrones GoF.	<ul style="list-style-type: none"> Desarrollar ejemplos de patrones. TAREA ACADÉMICA 1
	15	24	Clases Persistentes. Serialización en archivos XML	<ul style="list-style-type: none"> Desarrolla ejercicios usando clases persistentes usando patrón MVC.
		25	Programación con clases persistentes y bases de datos.	<ul style="list-style-type: none"> Desarrollar clases persistentes con base de datos usando MVC, DTO, DAO. AVANCE DE PROYECTO FINAL 3
	16	26	Programación orientado a objetos con bases de datos.	<ul style="list-style-type: none"> Desarrollar clases persistentes con base de datos usando MVC, DTO, DAO. TAREA ACADÉMICA 2
		27	Patrones de diseño básicos: singleton, adaptador, iterator.	<ul style="list-style-type: none"> Desarrollar programas utilizando

	17			patrones.
		28	Patrones de diseño básicos : builder, facade, observer, state.	<ul style="list-style-type: none"> • Desarrollar programas utilizando patrones.
	18	29	Entrega de Proyecto Final (4) con un avance de 100%.	<ul style="list-style-type: none"> • PROYECTO FINAL

ⁱ Debido a la coyuntura actual y acorde a la normativa, el curso se adaptará excepcionalmente a la educación no presencial, por tanto, los contenidos, actividades y cronograma serán adaptados por el o la docente para garantizar los aprendizajes señalados en el logro general de aprendizaje del curso.

ⁱⁱ Debido a la coyuntura actual y acorde a la normativa, el curso se adaptará excepcionalmente a la educación no presencial, por tanto, los contenidos, actividades y cronograma serán adaptados por el o la docente para garantizar los aprendizajes señalados en el logro general de aprendizaje del curso.