	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 1 de 9


IDENTIFICACIÓN

Nombre de la asignatura	Programación de computadores II		
Código de la asignatura	SS300		
Programa Académico	Ingeniería de Sistemas		
Créditos académicos	3		
Trabajo semanal del estudiante	Docencia directa: 4	Trabajo Independiente: 5	
Trabajo semestral del estudiante	144		
Pre-requisitos	Programación de Computadores I – SS200		
Co-requisitos			
Departamento oferente	Ingeniería de Sistemas		
Tipo de Asignatura	Teórico:	Teórico-Práctico: X	Práctico:
Naturaleza de la Asignatura	Habilitable:	No Habilitable: X	
	Validable:	No Validable: X	
	Homologable:	No Homologable: X	

PRESENTACIÓN

Esta asignatura precisa necesariamente de un conocimiento básico de los conceptos de programación, y el paradigma de la programación orientada a objetos está principalmente enfocada a mejorar el diseño de los propios programas. De esta manera, se introduce al alumno en la creación de programas desde un punto de vista más abstracto que la simple algorítmica básica, promoviendo un buen diseño, enfocado a resolver los aspectos más relevantes del problema, mientras lo específico queda oculto. De este modo, a diferencia de las asignaturas orientadas a la programación procedural, donde la sintaxis y la semántica del lenguaje son primordiales, esta asignatura se orienta hacia el diseño y estructura de los programas en su conjunto. Sin embargo, también se introduce y se usa el lenguaje Java para implementar los programas diseñados.

La organización de la asignatura sigue la siguiente estructura: Al principio de la asignatura se estudian los conceptos y principios fundamentales de la programación orientada a objetos, como la abstracción, la encapsulación y la reutilización. En una segunda parte de la asignatura, utilizando los conocimientos básicos anteriores, se introducen conceptos más avanzados como la herencia, el polimorfismo, la sobrecarga y las interfaces; estudiando especialmente cómo aplicarlos y utilizarlos en la construcción de programas orientados a objetos. En la última parte de la asignatura se aprende como aplicar todos los conceptos en su conjunto para resolver problemas complejos. Esto se desarrolla mediante el estudio del modelado de objetos y sus relaciones, así como la reutilización de código, y también el estudio de problemas concretos resueltos utilizando objetos.

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 2 de 9

JUSTIFICACIÓN

La Orientación a Objetos - OO, es un paradigma de programación que permite reducir la complejidad inherente al desarrollo de software, mediante la abstracción de datos, la división del programa en módulos manejables y la reutilización de código, permitiendo reducir el tiempo y esfuerzo en la creación de soluciones y desarrollo de pruebas, por lo cual, se ha consolidado a través de los años como una de las herramientas indispensables para la ejecución de proyectos empresariales de construcción de software, y una de las más utilizadas en los últimos desarrollos empresariales implementados.


La industria moderna del software tiene en la programación orientada a objetos toda una filosofía para conceptualizar y representar la realidad tangible e intangible, debido a la versatilidad que proporciona a los programadores y a la flexibilidad que demuestra para el desarrollo de sistemas de información. Es por esto, que en la actualidad la orientación a objetos se constituye en una herramienta fundamental para el diseño y desarrollo de aplicaciones para computador, y como evidencia de ello, el crecimiento exponencial del número de lenguajes programación que soportan la orientación a objetos que se ha dado desde sus orígenes.

Es tanta la importancia que ha tomado este paradigma de programación, que las universidades más importantes e influyentes del mundo incluyen dentro de su sistema curricular, como curso básico y fundamental, los conceptos de la orientación a objetos en etapas de iniciación a las ciencias de la computación, con la finalidad de aportar a los estudiantes y futuros ingenieros elementos innovadores en cuanto a métodos y estrategias adoptadas para la solución de problemas susceptibles de implementar en computadora mediante la concepción de algoritmos.

Por tanto, la posibilidad de ofrecer a los estudiantes, de manera complementaria a los conceptos de estructuras algorítmicas tradicionales, las técnicas y mecanismos de la programación orientada a objetos, permitirá garantizar, desde los procesos formativos en ciería, la formación de profesionales competentes, con dominio de las herramientas fundamentales requeridas por la actual industria del software.

OBJETIVO GENERAL

Lograr que el estudiante domine los conceptos y técnicas básicas del Paradigma de Programación Orientada a Objetos, y los aplique en el diseño y construcción de soluciones de software de mediana complejidad, para obtener como resultado aplicativos robustos, fiables, flexibles, mantenibles, escalables y bien documentados.

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
	PLAN DE ASIGNATURA	PÁG: 3 de 9

OBJETIVOS ESPECÍFICOS

- Comprender el concepto de paradigma de programación y las principales diferencias conceptuales entre los paradigmas imperativo, funcional, y orientado a objetos.
- Entender la estructura de un programa organizado con objetos.
- Diseñar las clases necesarias y construir los objetos requeridos que permitan dar solución a problemas computacionales en términos del paradigma orientado a objetos.
- Reconocer los principales elementos del Lenguaje Unificado de Modelado y representar mediante diagramas UML las clases y relaciones requeridas para construir soluciones de software dentro del paradigma orientado a objetos.
- Reconocer las características y sintaxis de un lenguaje de programación basado en el paradigma orientado a objetos.
- Desarrollar aplicaciones de mediana complejidad utilizando objetos, clases y demás mecanismos propios de la POO.
- Implementar programas utilizando un lenguaje de programación de alto nivel orientado a objetos (opcional Java).


COMPETENCIAS GENERALES Y ESPECÍFICAS

Competencias Generales

- ✓ Reconoce la importancia del trabajo en equipo, la permanente comunicación e interacción para el logro de las metas propuestas en el contexto.
- ✓ Responde con responsabilidad a las actividades asignadas para el logro de los objetivos propuestos en la asignatura.
- ✓ Toma decisiones previo análisis de ventajas y desventajas a las que estas pueden conducir y su incidencia en las metas propuestas en el contexto

Competencias Especificas

Aplica el Paradigma Orientado a Objetos para solucionar problemas del entorno, a través

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 4 de 9

de conceptos y técnicas de los pilares de la Programación Orientada a Objetos.

- Capacidad para solucionar problemas mediante el diseño y creación de algoritmos.
- Apropia los conceptos de la Programación Orientada a Objetos.
- Utiliza técnicas de modelado para la solución de problemas del entorno.
- Aplica la sintaxis de un lenguaje orientado a objetos.
- Usa un lenguaje de programación orientado a objetos para desarrollar programas que dan solución a problemas del entorno.

METODOLOGÍA

La asignatura será dirigida por el docente, teniendo en cuenta los temas establecidos por la dirección del programa, a través de:


Docencia Directa: Esta estrategia corresponde a clases presenciales, dirigidas por el docente, el cual explicará y profundizará las ideas y conceptos principales de cada tema, fomentando la investigación, participación e interés del estudiante, mediante el diseño y dirección de proyectos, talleres, debates, mesas redondas, sustentaciones y socialización de: lecturas autorreguladas, mapas conceptuales y ensayos.

Trabajo Independiente: Esta estrategia corresponde al autoaprendizaje por parte del estudiante. Para ello, debe documentarse y preparar los diferentes temas de la asignatura con anticipación; teniendo en cuenta el contenido suministrado por el docente, utilizando las diferentes fuentes bibliográficas.

Horas de Asesoría: Esta estrategia corresponde a la asesoría que debe brindar el docente a los estudiantes, sobre las tareas asignadas y en horas estipuladas independientemente de las horas de docencia directa.

ESTRATEGIAS METODOLÓGICAS

Entre las estrategias pedagógicas utilizadas por el docente para impartir la asignatura se

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 5 de 9


encuentran:

- **Talleres.** Esta estrategia metodológica fortalece el proceso de enseñanza- aprendizaje; el taller es una actividad práctica que promueve un espacio de reflexión y construcción del conocimiento; estos son previamente diseñados por los docentes con base a las competencias que el estudiante debe desarrollar en cada asignatura y publicados en espacios tales como: web sites, blogs, aula web o aula de clases. Las asignaturas de tipo teórico - práctico usan esta estrategia para promover el trabajo en equipo, consultas y profundización investigativa.
- **Mediaciones Virtuales.** El uso y apropiación de los tics se convierten en herramientas claves que son de apoyo al proceso de formación en el aula de clases, debido a que promueven en el estudiante la búsqueda permanente del conocimiento a través de herramientas como: plataformas virtuales- aula web, redes profesionales, sociales, web sites, aplicaciones en la nube, correo electrónico, foros y demás herramientas sincrónicas y asincrónicas que facilitan la interacción.
- **Visitas empresariales:** estas acercan al estudiante con aplicaciones y situaciones reales de la ingeniería de sistemas en los diferentes sectores productivos. Asociadas al conocimiento que el estudiante debe tener de su entorno, persiguiendo la construcción de pensamiento crítico y el aprendizaje significativo.
- **Proyecto de aula:** esta estrategia corresponde al desarrollo del proyecto guiado por el docente desde el inicio del semestre y donde el estudiante es el actor principal, quien debe identificar problemas del entorno y a través de aplicativos confiables contribuir a la optimización de los procesos.

CONTENIDO

Unidad 1. Introducción a la Programación Orientada a Objetos (POO)

- 1.1. Paradigmas de la programación
- 1.2. Características de la Programación Orientada a Objetos
- 1.3. Conceptos de Programación Orientada a Objetos
 - 1.3.1. Abstracción
 - 1.3.2. Encapsulamiento

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 6 de 9

- 1.3.3. Polimorfismo
- 1.3.4. Herencia
- 1.3.5. Modularidad
- 1.4. Clases y Objetos
- 1.5. Mensajes y Métodos

Unidad 2. Nociones del lenguaje de Programación Orientada a Objetos

- 2.1 Generalidades
- 2.2 Sintaxis básica
- 2.3 Comentarios, identificadores y palabras reservadas
- 2.4 Tipos datos primitivos
- 2.5 Variables y constantes
- 2.6 Operadores
- 2.7 Sentencias de control de flujo de ejecución
- 2.8 Enumerados
- 2.9 Arrays (Vectores, Matrices)


Unidad 3. Programación basada en objetos

- 3.1 Clases y Objetos (Estado, comportamiento e identidad)
- 3.2 Abstracción de objetos – Creación de Clases
- 3.3 Visibilidad de Clases y Objetos
 - 3.3.1 Vista pública de las Clases
 - 3.3.4 Vista privada de las Clase
- 3.4 Atributos y Métodos
 - 3.4.1 Miembros de Instancia y de Clase
 - 3.4.2 Método constructor
 - 3.4.3 Métodos Getter y Setters (encapsulamiento)
 - 3.4.4 Sobrecarga de métodos

Unidad 4. Lenguaje de Modelado Unificado (UML) y Relaciones entre clases

- 4.1 Introducción a UML
 - 4.2 Vistas UML
 - 4.3 Diagramas de casos de uso
 - 4.4 Diagramas de clases
 - 4.4.1 Representación de clases en UML
 - 4.4.2 Relación de asociación
 - 4.4.3 Relación de agregación
 - 4.4.4 Relación de composición
 - 4.4.5 Relación de herencia (Especialización/Generalización)
- Taller para el diseño de los diagramas de clases usando UML.

Unidad 5. Herencia, Polimorfismo e Interfaces

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 7 de 9

- 5.1 Herencia Simple
- 5.2 Herencia Múltiple
- 5.3 Polimorfismo
 - 5.3.1 Clases abstractas
 - 5.3.2 Métodos abstractos
- 5.4 Construcción e implementación de interfaces
- 5.5 Herencia Múltiple con Interfaces

Unidad 6. Programación con excepciones

- 6.1 Excepciones
- 6.2 Manejo de excepciones (bloque try – catch - finally)
 - 6.2.1 Excepciones verificadas y no verificadas
 - 6.2.2 Lanzar excepciones (clausula throws)
- 6.3 Creación de excepciones personalizadas

Unidad 7. Colecciones de Objetos

- 7.1 Arrays de objetos
- 7.2 Clase ArrayList
- 7.3 ArrayList de Objetos
 - 7.3.1 Operaciones sobre una colección de objetos
 - 7.3.2 Adicionar objetos
 - 7.3.3 Buscar objetos
 - 7.3.4 Ordenar Colecciones
 - 7.3.5 Remover objetos

Unidad 8. Modelo tres capas


- 8.1 Creación de paquetes
- 8.2 Visibilidad de clases y miembros
- 8.3 Utilización de paquetes

Unidad 9. Persistencia de datos

- 9.1. Archivos binarios
- 9.2. Archivos de texto
- 9.3. Serialización (objetos)
- 9.4 Operaciones CRUD sobre archivos

Unidad 10. Interfaz gráfica de usuario - GUI

- 10.1 Componentes gráficos básicos
- 10.2 Manejo de layouts
- 10.3 Gestión de eventos
- 10.4 Arquitectura de la interfaz gráfica

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 8 de 9

EVALUACIÓN

La calificación de la asignatura está organizada de la siguiente forma:

PARCIALES:

- **Primer parcial:** 30%(5% talleres, trabajos, 5% primera entrega del proyecto final y 20% parcial).
- **Segundo parcial:** 30%(5% talleres, trabajos y asistencia, 5% segunda entrega del proyecto final y parcial 20%.
- **Tercer parcial:** 40%(20% Parcial y 20% Trabajo final).


TRABAJOS

Asistencia a clases.
Ejercicios.
Exposiciones.
Talleres.

REFERENCIAS BIBLIOGRÁFICAS

Biblioteca:

- Cernura, A. *Análisis y diseño estructurado y orientado a objetos de sistemas informáticos*, Addison-Wesley (1996).
- Joyanes, L. *Programación orientada a objetos*, McGraw Hill (1998).
- Ceballos, F. *Java 2 interfaces gráficas y aplicaciones para internet*, Grupo AlfaOmega (2006).
- Joyanes, L. *Programación en Java 2: Algoritmos, estructuras de datos y programación orientada a objetos*, McGraw Hill (2002).
- Fuente. P. *UML para programadores java*, Pearson Educación (2009)
- Herbert, James. *El arte de programar en java*, McGraw Hill (2004).
- Joyanes, L. *Fundamentos de Programación: Algoritmos, estructura de datos y objetos*, McGraw Hill (2008)

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 9 de 9

- BRENTA, B. Programación orientada a objetos con java, PEARSON (2007).
- Bennet, S. Análisis y diseño orientado a objetos de sistemas usando UML, McGraw Hill (2007).
- Deitel y Deitel. Como programar en Java, Prentice Hall Hispanoamericana (1998).

Base de datos digitales:

e-libro

Jaramillo Valbuena, Sonia, Cardona Torres, Sergio Augusto, and Hernández Rodríguez, Leonardo Alonso. Programación orientada a objetos. Bogotá, CO: Ediciones Elizcom, 2010. ProQuest ebrary. Web. 14 June 2017.

Flórez Fernández, Héctor Arturo. Programación orientada a objetos usando java. Bogotá, CO: Ecoe Ediciones, 2012. ProQuest ebrary. Web. 14 June 2017.

García Llinás, Luis Fernando. Todo lo básico que debería saber: sobre programación orientada a objetos en Java. Bogotá, CO: Ediciones de la U, 2010. ProQuest ebrary. Web. 14 June 2017.

Vélez Serrano, José, Peña Abril, Alberto, and Gortazar Bellas, Patxi. Diseñar y programar, todo es empezar: una introducción a la Programación Orientada a Objetos usando UML y Java. Madrid, ES: Dykinson, 2011. ProQuest ebrary. Web. 14 June 2017.

Moreno Pérez, Juan Carlos. Programación. Madrid, ES: RA-MA Editorial, 2014. ProQuest ebrary. Web. 14 June 2017.