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

IDENTIFICACIÓN			
Nombre de la asignatura	PROGRAMACION DE COMPUTADRES III		
Código de la asignatura	SS404		
Programa Académico	Ingeniería en Sistemas		
Créditos académicos	4		
Trabajo semanal del estudiante	Docencia directa: 4	Trabajo Independiente: 5	
Trabajo semestral del estudiante	144		
Pre-requisitos	PROGRAMACION DE COMPUTADRES II		
Co-requisitos			
Departamento oferente	Ingeniería en sistemas		
Tipo de Asignatura	Teórico:	Teórico-Práctico: X	Práctico:
Naturaleza de la Asignatura	Habilitable:		No Habilitable:
	Validable:		No Validable:
	Homologable: X		No Homologable:
PRESENTACIÓN			
<p>Asignatura teórica practica la cual pretende que el estudiante refuerce sus conocimientos, habilidades y destrezas en la resolución de problemas de tipo algorítmicos a través de un lenguaje de programación y un entorno de desarrollo específico, que le permita construir programas para un entorno de producción.</p>			
JUSTIFICACIÓN			
<p>La resolución de problemas a través del paradigma de orientación a objeto, permiten que al estudiante modele la realidad para obtener soluciones ajustadas a las necesidades que el entorno requiere.</p> <p>El estudiante de ingeniería de sistemas debe poseer bases sólidas en el manejo de herramientas computacionales, que le permitan ser competitivo en una sociedad cada vez más exigente. El curso está encaminado a que el estudiante tenga la capacidad de desarrollar aplicaciones que modelen la realidad, tanto de situaciones hipotéticas como reales en entornos de desarrollo o de producción con paradigma orientado a objetos, aplicando principios y buenas prácticas de programación, teniendo en cuenta criterios de usabilidad y rendimiento.</p> <p>Además, proporcionara al estudiante bases y conceptos necesarios para entender los criterios de alcance y complejidad de las aplicaciones con el fin de estimar mejor el esfuerzo necesario para desarrollarlas.</p>			
OBJETIVO GENERAL			

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 2 de 8


Desarrollar soluciones de software que modelen la realidad ajustándose a las necesidades que el entorno requiere, aplicando los principios y buenas prácticas de la programación orientada a Objetos.

OBJETIVOS ESPECÍFICOS

1. Adquirir habilidad en el uso del lenguaje y el entorno de desarrollo seleccionado.
2. Diseñar e implementar algoritmos que dividan la complejidad usando técnicas orientadas a objetos.
3. Conocer e implementar objetos de uso común en las aplicaciones con interfaz gráfica de usuario GUI.
4. Usar técnicas orientadas a objetos para crear clases propias o extender las propiedades de una clase existente.
5. Conocer e implementar soluciones comunes a problemas comunes.
6. Elaborar programas usando Clases y objetos de bibliotecas y propios que modelen la realidad.
7. Crear aplicaciones con persistencia de datos.

COMPETENCIAS GENERALES Y ESPECÍFICAS

PROPOSITIVAS	Capacidad para organizar y planear Capacidad para trabajar autónomamente Capacidad para tomar decisiones Capacidad para solucionar problemas Capacidad de ejercer liderazgo Capacidad para generar nuevas ideas (creatividad)
CONVIVENCIA	Capacidad para trabajar en equipo e interdisciplinariamente Capacidad de construir consensos Capacidad para el desarrollo del pensamiento crítico, abierto y reflexivo Capacidad de autorreflexión y autocrítica Capacidad para actuar con ética, disciplina y responsabilidad
APRENDIZAJE	Capacidad para aprender Capacidad para analizar y sintetizar Capacidad para adaptarse a los cambios o nuevas situaciones
COMUNICATIVAS	Capacidad para la comunicación oral y escrita Capacidad para aplicar las habilidades básicas computacionales o en informática Capacidad para buscar, manejar y usar información

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 3 de 8

Específicas:

1. Diseñar e implementar algoritmos utilizando la estrategia de divide y vencerás para la construcción de programas orientados a objetos
2. Conocer y usar algoritmos utilizados históricamente para búsqueda y clasificación.
3. Resolver problemas utilizando algoritmos de ordenación eficientes.
4. Reconocer y usar objetos comunes en aplicaciones basadas en formularios
5. Construir aplicaciones que consuman datos; usando el paradigma de programación orientada a objetos, reciclando objetos de las librerías y creando objetos propios.
6. Construir aplicaciones que usen más de un proceso (multihilos)
7. Diseñar e implementar soluciones bajo la arquitectura de tres capas

Genéricas:

1. Saber cómo encontrar bibliotecas de soporte para la construcción y mantenimiento de aplicaciones
2. Utilizar clases de biblioteca y algoritmos comunes disponibles en el código de las aplicaciones
3. Conocer las ventajas y desventajas en la elección del algoritmo apropiado para realizar tareas comunes.
4. Destreza en la refactorización de métodos y clases


Usar más de un proceso en las aplicaciones

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.


	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 4 de 8

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 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 las 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.

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 5 de 8

CONTENIDO

UNIDAD I: Buenas Prácticas de POO

1. Introducción al Lenguaje y Framework.

- 1.1. Framework de .Net
- 1.2. Tipos de Datos
- 1.3. Declaraciones de Variables
- 1.4. Arreglos
- 1.5. Operadores Aritméticos
- 1.6. Operadores Lógicos
- 1.7. Comparadores
- 1.8. Aplicación de Consola
- 1.9. Condicionales
- 1.10. Instrucciones repetitivas
- 1.11. Colección de Objetos

2. Fundamentos de POO y Principios

- 2.1. Por qué la POO
- 2.2. Que son los principios de Programación
- 2.3. Porque debo utilizar los principios
- 2.4. Que beneficios obtengo
- 2.5. Bajo acoplamiento
- 2.6. Alta Cohesión

3. Código Limpio

- 3.1. Nombres con sentido
- 3.2. Tamaño de las funciones
- 3.3. Como usar comentarios


4. Programación en Capas

- 4.1. Crear proyectos de bibliotecas de Clases de enlaces dinámico
- 4.2. Capa de presentación
- 4.3. Capa de Lógica
- 4.4. Capa de Acceso a Datos
- 4.5. Clases del dominio y clases de fabricación pura


UNIDAD II: CAPA DE LOGICA

5. Principio de Diseño Parte I

- 5.1. Responsabilidad única
- 5.2. Principio Open-Close
 - 5.2.1. Herencia y Composición como mecanismos de extensión de código.
 - 5.2.2. Composición vs Herencia.

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 6 de 8

- 5.3. Principio de Liskov
- 5.3.1. Aplicación de polimorfismo.
- 5.3.2. Interfaces vs clases abstractas
- 5.4. Principio Segregación de interfaces
- 5.5. Inyección de dependencia
- 5.5.1. Agregación.
- 5. Clases del dominio y clases de fabricación pura**
- 7. Modelando el mundo el mundo real con objetos**
- 7.1. Asociación
- 7.1.1. Unos a muchos
- 7.1.2. Uno a Uno
- 7.1.3. Muchos a muchos
- 7.2. Tipos de Asociación
- 7.2.1. Composición
- 7.2.2. Herencia
- 7.2.3. Agregación
- 7.2.4. Dependencia
- UNIDAD III: CAPA DE PRESENTACIÓN**
- 8. GUI**
- 9. Formularios**
- 9.3. Objetos contenedores
- 9.4. Objetos comunes
- 9.5. Objetos Menús, Barras de herramientas y Barras de estado.
- 10. Tipos de Formularios**
- 10.1 Modal
- 10.2 MDI
- 11. Principales eventos de GUI**
- 12. Principios de Diseño de Interfaz de Usuarios**
- 12.1. Busca la consistencia
- 12.2. Permite a los usuarios frecuentes utilizar atajos
- 12.3. Ofrece respuestas a las acciones
- 12.4. Diseña acciones secuenciales
- 12.5. Gestión de errores sencilla
- 12.6. Fácil reversión de las acciones.
- 12.7. Apoya al usuario con una interfaz amigable.
- 12.8. Reduce la carga de memoria a corto plazo

	UNIVERSIDAD POPULAR DEL CESAR	CODIGO: 201-300-PRO05-FOR01
		VERSIÓN: 1
PLAN DE ASIGNATURA		PÁG: 7 de 8

13. Otros principios de programación

- 13.1. KISS,
- 13.2. DRY
- 13.3. YAGNIS
- 14. Validación de Datos en las Interfaces de Usuario

UNIDAD IV: Capa de Acceso a Datos.

15. Introducción a las bases de datos:

- 15.1. Base de datos relacionales
- 15.2. Tablas, Filas y Campos
- 15.3. Creación de una base de datos
- 15.4. Comando de manipulación de datos (DML) (Insert, Update, Select, Delete)

16. Persistencia de Objetos a Base de Datos

17. Patrones de Acceso a datos

- 17.1. Active Record
- 17.2. DAO (Data Access Object)
- 17.3. DTO (Data Transfer Object)
- 17.4. Repositorio

18. Mapear Objetos a Base de Datos.

19. Manejo de Acceso a Datos

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.

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

REFERENCIAS BIBLIOGRÁFICAS

LARMAN, C. (2003). UML Y PATRONES: INTRODUCCION AL ANALISIS Y DISEÑO ORIENTADO A OBJETO. Pearson Hall.

ROBERT C. MARTIN (2009). CÓDIGO LIMPIO: MANUAL DE ESTILO PARA EL DESARROLLO ÁGIL DE SOFTWARE. Pearson Hall

REFERENCIAS EN LÍNEA:

Base de conocimiento en la web <http://msdn.microsoft.com/es-co/default.aspx>
 Microsoft Virtual Academy <https://mva.microsoft.com/> (recursos normalmente en inglés)

Ceballos Sierra, Francisco Javier. Microsoft C#: lenguaje y aplicaciones (2a. ed.). Madrid, ES: RA-MA Editorial, 2007. ProQuest ebrary. Web. 17 June 2017.

Ceballos Sierra, Francisco Javier. Enciclopedia de Microsoft Visual C#: interfaces gráficas y aplicaciones para Internet con Windows Forms y ASP.NET (4a. ed.). Madrid, ES: RA-MA Editorial, 2012. ProQuest ebrary. Web. 17 June 2017.

Introducción a .NET. Barcelona, ES: Editorial UOC, 2010. ProQuest ebrary. Web. 17 June 2017.

Ceballos Sierra, Francisco Javier. Visual Basic: interfaces gráficas y aplicaciones para Internet con WPF, WCF y Silverlight. Madrid, ES: RA-MA Editorial, 2012. ProQuest ebrary. Web. 17 June 2017.

REFERENCIAS FÍSICAS:

Enciclopedia de Microsoft Visual C#.Net, 3 edición, Alfa & Omega, Francisco Javier Ceballos

Tom Archer(2001), C# A Fondo, McGraw-Hill