

PROGRAMA DE ASIGNATURA

I.- DATOS GENERALES

Nombre de la Carrera o Programa: Ingeniería Informática

Nombre de la Asignatura: Desarrollo de Software

Departamento: Departamento de Ingeniería del Software

Régimen: Semestral

Número de Unidades Crédito: 5

Ubicación en el plan de estudios: Octavo Semestre

Requisitos:
 Metodología del Software + Sistemas de Bases de Datos II

Asignaturas a las que aporta:
 Pasantía, Gestión de Proyectos de Software, Trabajo Instrumental de Grado

Tipo de asignatura:

Obligatoria: X	Electiva:
----------------	-----------

Horas semanales:

Teoría: 2	Práctica: 2	Laboratorio: 0
-----------	-------------	----------------

Vigente desde: Octubre 2015

II.- JUSTIFICACIÓN

La unidad curricular Desarrollo de Software contribuye a la formación de Ingenieros en Informática éticos competentes en el desarrollo de software en diferentes plataformas, más específicamente se desarrolla un sistema de información utilizando un manejador de base de datos objeto-relacional (OR) para el almacenamiento de los datos y un sistema n-capas programado utilizando técnicas del paradigma de programación orientado a objetos. Afianza y amplía las competencias adquiridas por los estudiantes en unidades curriculares previas como Algoritmos y Programación III, Ingeniería del Software, Sistemas de Bases de Datos I, Sistemas de Bases de Datos II y Metodología del Software, especialmente en lo que se refiere al aprendizaje crítico, reflexivo, colaborativo y autónomo favoreciendo el aumento de las actitudes de liderazgo, compromiso con el bienestar del otro y compromiso con la excelencia. El proyecto que se diseña e implementa durante el semestre es de aplicación práctica de la vida real, ayudando al estudiante a enriquecer su currículum profesional aún antes de graduarse.

III.- CONTRIBUCIÓN DE LA ASIGNATURA AL DESARROLLO DE LAS COMPETENCIAS

Competencia General 1 (CG1): Aprender a aprender con calidad

Unidad de Competencia 1 (CG1 - U1):

Abstrae, analiza y sintetiza información

Criterios de desempeño de la U1:

1. Identifica elementos comunes en diferentes situaciones o contextos
2. Resume información de forma clara y ordenada
3. Integra los elementos de forma coherente

Unidad de Competencia 2 (CG1 - U2):

Aplica los conocimientos en la práctica

Criterios de desempeño de la U2:

1. Selecciona la información que resulta relevante para resolver una situación
2. Implementa el proceso a seguir para alcanzar los objetivos mediante acciones, recursos y tiempo disponible

Unidad de Competencia 3 (CG1 - U3):

Identifica, plantea y resuelve problemas

Criterios de desempeño de la U3:

1. Reconoce diferencias entre una situación actual y la deseada

Unidad de Competencia 4 (CG1 - U4):

Demuestra conocimiento sobre su área de estudio y profesión

Criterios de desempeño de la U4:

1. Aplica con fluidez la terminología del área de estudio y profesión
2. Aplica los procedimientos de la disciplina para resolver problemas y aportar soluciones

Unidad de Competencia 5 (CG1 - U5):

Trabaja con altos estándares de calidad

Criterios de desempeño de la U5:

1. Actúa conforme a las normas y exigencias que denotan la calidad de su actuación

Competencia General 2 (CG2): Aprender a trabajar con el otro

Unidad de Competencia 1 (CG2 - U1):

Participa y trabaja en equipo

Criterios de desempeño de la U1:

1. Realiza las tareas establecidas por el equipo

Competencia Profesional Básica 1 (CPB1): Formula Proyectos de Ingeniería

Unidad de Competencia 1 (CPB1 - U1):

Cumple con el código de ética profesional y el marco legal vigente

Criterios de desempeño de la U1:

1. Aplica el código de ética en su ambiente profesional

Competencia Profesional Específica 1 (CPE1): Gestiona proyectos informáticos

Unidad de Competencia 1 (CPE1 - U1):

Documenta técnica y funcionalmente un sistema informático

Criterios de desempeño de la U1:

1. Elabora el manual de referencia técnica de un sistema informático

Competencia Profesional Específica 2 (CPE2): Desarrolla Software de aplicación

Unidad de Competencia 1 (CPE2 - U1):

Diseña e implementa algoritmos robustos y eficientes de forma modular

Criterios de desempeño de la U1:

1. Utiliza la abstracción para diseñar de manera eficiente la solución de un problema
2. Implementa la solución algorítmica
3. Identifica y resuelve las situaciones excepcionales de un programa para lograr una solución robusta

III.- CONTRIBUCIÓN DE LA ASIGNATURA AL DESARROLLO DE LAS COMPETENCIAS

<p>Unidad de Competencia 2 (CPE2 - U2): Analiza las necesidades de los usuarios, diseña e implementa el software de aplicación sobre arquitecturas centralizadas o distribuidas</p>	<p>Criterios de desempeño de la U2: 1. Diseña la solución planteada 2. Implementa la solución planteada</p>
<p>Unidad de Competencia 3 (CPE2 - U3): Diseña y ejecuta planes de pruebas de los sistemas informáticos</p>	<p>Criterios de desempeño de la U3: 1. Define el plan de pruebas funcionales y técnicas 2. Aplica el plan de pruebas</p>

IV.- UNIDADES TEMÁTICAS

UNIDADES	TEMAS
1. Requisitos del Software	1.1. Repaso de técnicas para el levantamiento de requisitos del software 1.2. Repaso de casos de uso y diagramas de casos de uso (UML)
2. Análisis y Diseño de Software Orientado a Objetos	2.1. Diseño orientado a objetos 2.2. UML para diseño 2.3. Modelo del Dominio 2.4. Diseño guiado por el dominio (DDD) 2.5. Fundamentos de Arquitectura 2.6. Componentes 2.7. Diagrama de componentes (UML) 2.8. Patrones de diseño
3. Construcción de Software Orientado a Objetos	3.1. Estrategias para la construcción y documentación del código 3.2. Manejo de excepciones y tolerancia a fallos en aplicaciones multicapas 3.3. Depuración y uso de herramientas de depuración 3.4. Uso de herramientas para la construcción del software
4. Pruebas de Sistemas Informáticos	4.1. Fundamentos de pruebas del software y tipos de pruebas 4.2. Estrategias y técnicas para la automatización de pruebas 4.3. Desarrollo guiado por pruebas 4.4. Patrones para la implementación de pruebas unitarias
5. Implantación de Sistemas Informáticos	5.1. Control de versiones 5.2. Tipos de control de versiones (centralizados vs. distribuidos) 5.3. Generación de entregas (software releases) 5.4. Diagrama de despliegue (UML) 5.5. Documentación técnica 5.6. Preparación de ambientes de producción
6. Gerencia de Proyectos	6.1. Fundamentos y conceptos 6.2. Planificación de proyectos de software 6.3. Estimación de costos

V.- ESTRATEGIAS DE ENSEÑANZA Y DE APRENDIZAJE

Consultas en diferentes fuentes: digitales, impresas, bibliográficas; Wiki; Preguntas generadoras y preguntas guías; Discusión; Resolución de ejercicios y problemas; Método de proyectos; Video; Conferencia; Aprendizaje basado en problemas

VI.- ESTRATEGIAS DE EVALUACIÓN

Observación, Técnica de la pregunta reflexiva, Pruebas escritas y orales, Resolución de ejercicios y problemas, Proyectos, Pruebas objetivas, Pruebas de libro abierto, Rúbrica

VII.- REFERENCIAS BIBLIOGRÁFICAS

Textos:

1. Meyer, B. *Object-oriented software construction*. Prentice Hall.
2. Gamma, E., Booch, G., Johnson, R., Vlissides, J. M. & Helm, R. *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley Professional.
3. Braude, E. J. (2006). *Software design: From programming to architecture*. John Wiley & Sons.
4. Larman, C. (2004). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development* (3ra ed.). Estados Unidos: Prentice Hall.
5. Meszaros, G. & Fowler, M. (2007). *XUnit test patterns: Refactoring test code* (3ra ed.). Estados Unidos: Addison-Wesley Educational Publishers.
6. McConnell, S. (2004). *Code complete: [a practical handbook of software construction]* (2da ed.). Estados Unidos: Microsoft Press.
7. Bass, L., Clements, P. & Kazman, R. (2012). *Software architecture in practice* (3ra ed.). Estados Unidos: Addison-Wesley Educational Publishers.
8. Pressman, Roger S. *Ingeniería de Software. Un enfoque Práctico*. McGraw-Hill.

Web:

1. Object Management Group: <http://www.omg.org>