

ASIGNATURA DE PROGRAMACIÓN DE APLICACIONES

UNIDADES DE APRENDIZAJE

1. Competencias	Dirigir proyectos de tecnologías de información (T.I.) para contribuir a la productividad y logro de los objetivos estratégicos de las organizaciones utilizando las metodologías apropiadas. Evaluar sistemas de tecnologías de información (T.I.) para establecer acciones de mejora e innovación en las organizaciones mediante el uso de metodologías para auditoría.
2. Cuatrimestre	Séptimo
3. Horas Teóricas	31
4. Horas Prácticas	44
5. Horas Totales	75
6. Horas Totales por Semana Cuatrimestre	5
7. Objetivo de Aprendizaje	El alumno empleará el paradigma de la programación Orientada a Objetos para el desarrollo de sistemas de información y su seguridad a nivel avanzado.

Unidades de Aprendizaje	Horas		
	Teóricas	Prácticas	Totales
I. Principios básicos de la programación orientada a objetos.	5	5	10
II. Conceptos avanzados de la programación orientada a objetos.	5	10	15
III. Patrones de diseño.	15	25	40
IV. Seguridad en el desarrollo de aplicaciones.	6	4	10
Totales	31	44	75

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

UNIDADES DE APRENDIZAJE

1. Unidad de Aprendizaje	I. Principios básicos de la programación orientada a objetos.
2. Horas Teóricas	5
3. Horas Prácticas	5
4. Horas Totales	10
5. Objetivo de la Unidad de Aprendizaje	El alumno programará aplicaciones Orientadas a Objetos para satisfacer las necesidades básicas de la empresa.

Temas	Saber	Saber hacer	Ser
Paradigma de POO, Clases y Objetos, Atributos, Métodos y Herencia.	Definir los conceptos de clase, objetos, atributos, métodos y herencia. Reconocer las buenas prácticas de programación.	Formular programas empleando las clases, objetos, atributos, métodos y herencia.	Analítico Ordenado Sistemático Objetivo Ético Coherente Proactivo Asertivo
Agregación y asociación.	Definir los conceptos de Agregación y Asociación.	Emplear el paradigma de POO en una aplicación solicitada.	Analítico Ordenado Sistemático

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
Elaborará una aplicación que contenga: • Clases, objetos, atributos, métodos y herencia. • Relaciones de agregación y/o asociación documentada en un reporte que incluya: • Plantilla personalizada para evitar errores de sintaxis en la codificación. • Código fuente de la aplicación.	1. Identificar el paradigma de POO. 2. Identificar los conceptos de: clases, objetos, Atributo, herencia. 3. Codificar la aplicación utilizando los conceptos de POO. 4. Verificar el funcionamiento de la aplicación.	Ejecución de tareas Listas de verificación

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
Ejercicios prácticos Estudios de Casos Aprendizaje basado en Proyectos	Cañón Equipo de cómputo Software de programación Pintarrón Plumones

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

UNIDADES DE APRENDIZAJE

1. Unidad de Aprendizaje	II. Conceptos avanzados de la programación orientada a objetos.
2. Horas Teóricas	5
3. Horas Prácticas	10
4. Horas Totales	15
5. Objetivo de la Unidad de Aprendizaje	El alumno programará aplicaciones Orientadas a Objetos mediante los conceptos avanzados de este paradigma, para integrar la información en los sistemas.

Temas	Saber	Saber hacer	Ser
Polimorfismo.	Identificar el concepto de Polimorfismo en el POO.	Desarrollar aplicaciones empleando el concepto de Polimorfismo en un programa.	Analítico Ordenado Sistemático Objetivo Ético Coherente Proactivo Planificador Creativo Innovador
Clases Abstractas.	Identificar el concepto de clases abstractas en el POO.	Desarrollar aplicaciones empleando el concepto de clases abstractas.	Analítico Ordenado Sistemático Objetivo Ético Coherente Proactivo Planificador Creativo Innovador

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

Temas	Saber	Saber hacer	Ser
Interfaces.	Identificar el concepto de interfaces en el POO.	Programar una aplicación empleando conceptos avanzados de programación Orientada a Objetos.	Analítico Ordenado Sistemático Objetivo Coherente Proactivo Planificador Creativo Innovador

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
Elaborará una aplicación que contenga: • Polimorfismo. • Clases abstractas. • Interfaces. • Ventajas y desventajas de cada caso. La aplicación deberá estar documentada en un reporte que incluya: • Plantilla personalizada para evitar errores de sintaxis en la codificación. • Código fuente de la aplicación.	1. Identificar el Polimorfismo, Clases Abstractas e Interfaces. 2. Codificar una aplicación empleando dichos conceptos. 3. Verificar el funcionamiento de la aplicación.	Ejecución de Tareas Listas de verificación

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
Ejercicios prácticos Estudios de Casos Aprendizaje basado en Proyectos	Cañón Equipo de cómputo Software de programación Pintarrón Plumones

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

UNIDADES DE APRENDIZAJE

1. Unidad de Aprendizaje	III. Patrones de diseño.
2. Horas Teóricas	15
3. Horas Prácticas	25
4. Horas Totales	40
5. Objetivo de la Unidad de Aprendizaje	El alumno desarrollará aplicaciones utilizando patrones de diseño para optimizar el desempeño de la solución.

Temas	Saber	Saber hacer	Ser
Conceptos básicos de un patrón de diseño.	Identificar los conceptos de un patrón de diseño.		Analítico Ordenado Sistemático Objetivo
Patrones de diseño.	Identificar las características de los diferentes patrones de diseño existentes: Singleton. Refactory. Proxy. MVC.	Desarrollar aplicaciones empleando diferentes patrones de diseño.	Analítico Ordenado Sistemático Objetivo Coherente Proactivo Planificador Creativo Innovador Comprometido Responsable

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
Elaborará una aplicación utilizando un patrón de diseño, documentada en un reporte que incluya: • Plantilla personalizada para evitar errores de sintaxis en la codificación. • Código fuente de la aplicación. • Justificación del patrón de diseño empleado.	1. Identificar las características de los diferentes patrones de diseño. 2. Seleccionar el patrón de diseño adecuado a la organización. 3. Codificar los patrones de diseño en una aplicación. 4. Verificar el funcionamiento de la aplicación.	Ejecución de Tareas Listas de verificación

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
Ejercicios prácticos Estudios de Casos Aprendizaje basado en Proyectos	Cañón Equipo de cómputo Software de programación Pintarrón Plumones

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACION DE APLICACIONES

UNIDADES DE APRENDIZAJE

1. Unidad de Aprendizaje	IV. Seguridad en el desarrollo de aplicaciones.
2. Horas Teóricas	6
3. Horas Prácticas	4
4. Horas Totales	10
5. Objetivo de la Unidad de Aprendizaje	El alumno utilizará las mejores prácticas en el campo de la Seguridad de Software para el desarrollo de aplicaciones confiables.

Temas	Saber	Saber hacer	Ser
Seguridad en Informática.	Definir seguridad en informática aplicada a la programación.		Ordenado Sistemático Objetivo
Mejores Prácticas de seguridad del software.	Identificar las mejores prácticas en la seguridad del software en el desarrollo de aplicaciones.	Desarrollar aplicaciones empleando mejores prácticas de seguridad en la generación de aplicaciones confiables.	Analítico Ordenado Sistemático Objetivo Coherente Proactivo Planificador Creativo Innovador Organizado Responsable Disciplinado Comprometido Ético

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACION DE APLICACIONES

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
Elaborará una aplicación utilizando procedimientos de seguridad, documentada en un reporte que incluya: • Plantilla personalizada para evitar errores de sintaxis en la codificación. • Código fuente de la aplicación. • Justificación de las mejores prácticas empleadas.	1. Reconocer la importancia de la seguridad informática aplicada a la programación. 2. Enlistar las mejores prácticas en la seguridad del software. 3. Codificar una aplicación empleando las mejores prácticas en la seguridad del software mencionadas. 4. Verificar el funcionamiento de la aplicación.	Ejecución de Tareas Listas de verificación

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
Ejercicios prácticos Estudios de Casos Aprendizaje basado en Proyectos	Cañón Equipo de cómputo Software de programación Pintarrón Plumones

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

CAPACIDADES DERIVADAS DE LAS COMPETENCIAS PROFESIONALES A LAS QUE CONTRIBUYE LA ASIGNATURA

Capacidad	Criterios de Desempeño
Estructurar aplicaciones Web avanzadas móviles y de comercio electrónico basado en métodos de ingeniería de software y web, con bases de datos para garantizar la calidad del proceso de desarrollo.	Genera documentos de especificación de requerimientos conforme a los estándares y metodologías establecidas para ello. Genera el análisis y modelado de la aplicación de acuerdo a los requerimientos con base en los estándares y metodologías (Patrones de diseño, Ingeniería de Software e Ingeniería Web). Genera la aplicación con base en el modelado previamente establecido. Ejecuta plan de pruebas para verificar funcionalidad. Documenta los resultados.

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	

PROGRAMACIÓN DE APLICACIONES

FUENTES BIBLIOGRÁFICAS

Autor	Año	Título del Documento	Ciudad	País	Editorial
Erich Gamma	(2008)	<i>Patrones de Diseño</i>	Madrid	España	Addison Wesley
Garrido, José M.	(2003)	<i>Object-Oriented Programming (From Problem Solving to JAVA) (Programming Series)</i>	San Jose California	USA	Charles River Media
James W. Cooper	(2002)	<i>Introduction to Design Patterns in C#.</i>	San Jose California	USA	Addison-Wesley Professional
Steven John Metsker	(2004)	<i>Design Patterns in C#</i>	San Jose California	USA	Addison-Wesley Professional

REFERENCIAS (INTERNET)

Autor	Fecha de creación	Título del Documento	Consultado	Referencia
Yasar, Preuveneers, Berbers	(2008, diciembre 24)	<i>Best Practices for software security: An overview.</i>	31 de Marzo de 2009.	http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=4777689&arnumber=4777730&count=119&index=40

ELABORÓ:	Comité de Directores de la Carrera de Ing. en TIC	REVISÓ:	Comisión de Rectores de Fortalecimiento del SUT	
APROBÓ:	C. G. U. T.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2009	