



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Julio 05, 2022				
Carrera:	Ingeniería Mecatrónica	Asignatura:	Programación avanzada		
Academia:	Electrónica / Control	Clave:	19SME06		
Módulo formativo:	Programación	Seriación:	-		
Tipo de curso:	Presencial	Prerrequisito:	-		
Semestre:	Tercero	Créditos:	5.62	Horas semestre:	90 horas
Teoría:	2 horas	Práctica:	3 horas	Trabajo indpt.:	0 horas
				Total x semana:	5 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
OE1	El egresado solucionará problemas del entorno laboral en el que se desempeñe, mediante el uso de conocimientos técnicos adquiridos para la identificación, desarrollo innovador, aplicación y control de las posibles soluciones, utilizando sus habilidades en mecánica, electrónica, control y automatización para dar el resultado adecuado según las condiciones del problema.	El egresado aplicará las técnicas y metodologías para la identificación de problemas referentes a su entorno laboral, proponiendo soluciones creativas e innovadoras para los mismos.	% de alumnos que implementan diversidad de técnicas y metodologías para identificar problemas en su entorno laboral.
OE2	El egresado diseñará, mejorará o mantendrá de forma eficiente y sustentable equipos que cubran adecuadamente las diferentes necesidades del ámbito laboral, utilizando sus competencias técnicas de diseño, con sus conocimientos de materiales, control y procesos para lograr la mejor solución innovadora de la necesidad planteada.	El egresado fundamentará documentalmente la solución a problemas, desde la identificación hasta su resolución.	% de egresados que diseñan, mejoran o dan mantenimiento a equipos.
OE3	El egresado generará relaciones interpersonales y profesionales de otras áreas, para desarrollar habilidades técnicas, administrativas y colaborativas en el desarrollo de proyectos mecatrónicos.	El egresado desarrollará canales de comunicación y de gestión con departamentos y áreas relacionadas con los proyectos que lidera y coordina.	% de egresados que participan en más de un departamento y/o área por proyecto con las que se relaciona.



Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
AE1	Identificar y resolver problemas en el campo de la mecatrónica aplicando los principios de las ciencias básicas como la matemáticas y física, así como otras ciencias de la ingeniería.	- Identificar, desarrollar e implementar soluciones de software para las necesidades tanto comunitarias, comerciales e industriales aplicando la metodología de programación orientada a objetos.	<p>REPASO DE PROGRAMACIÓN ORIENTADA A OBJETOS</p> <ol style="list-style-type: none"> 1. Creación de Clases y Objetos 2. Alcance de las clases (Scope) 3. Métodos 4. Composición 5. Ámbito de las variables 6. Paquetes de clases 7. Modificadores de acceso (default, public, private, protected) 8. Herencia 9. Polimorfismo 10. Clases abstractas 11. Qué es una excepción 12. Manejo de excepciones (try/catch/finally) 13. Métodos que lanzan excepciones explícitamente <p>DISEÑO DE INTERFAZ GRÁFICA</p> <ol style="list-style-type: none"> 1. Definiciones: <ol style="list-style-type: none"> 1.1. Frame 1.2. Contenedores 1.3. Componentes 1.4. Eventos 2. Etiquetas 3. Botones 4. Campos de Texto 5. Áreas de Texto



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			<ul style="list-style-type: none"> 6. Listas combinadas 7. CheckBox 8. RadioButton 9. Scroll bars 10. Menús y Submenús 11. Ventanas emergentes 12. EVENTOS <ul style="list-style-type: none"> 12.1. Eventos de botón 12.2. Eventos de listas combinadas 12.3. Eventos de checkbox y radiobutton 12.4. Eventos de menús 12.5. Eventos de ratón 12.6. Eventos de teclado 12.7. Eventos de foco FUNDAMENTOS DE BASES DE DATOS <ul style="list-style-type: none"> 1. Estructura de una base de datos 2. Creación de una base de datos 3. Lenguaje de consulta de datos SQL (Structured Query Language) 4. Consultas, actualizaciones, modificaciones, borrado de registros HILOS <ul style="list-style-type: none"> 1. Conceptos básicos sobre hilos 2. Ejecución en segundo plano 3. Creación de un hilo 4. Estado y control de un hilo 5. Sincronización de hilos



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			PROGRAMACIÓN PARA REDES 1. Sockets 2. Aplicaciones Cliente-Servidor 3. Clientes múltiples
AE3	Desarrollar procesos y productos industriales desde un enfoque mecánico, electrónico, robótico, automatización y control, utilizando el juicio ingenieril para establecer conclusiones.	- Resolver un conjunto de prácticas de laboratorio para aplicar los conocimientos y habilidades de la programación orientada a objetos en el desarrollo de soluciones prácticas.	REPASO DE PROGRAMACIÓN ORIENTADA A OBJETOS 1. Creación de Clases y Objetos 2. Alcance de las clases (Scope) 3. Métodos 4. Composición 5. Ámbito de las variables 6. Paquetes de clases 7. Modificadores de acceso (default, public, private, protected) 8. Herencia 9. Polimorfismo 10. Clases abstractas 11. Qué es una excepción 12. Manejo de excepciones (try/catch/finally) 13. Métodos que lanzan excepciones explícitamente DISEÑO DE INTERFAZ GRÁFICA 1. Definiciones: 1.1. Frame 1.2. Contenedores 1.3. Componentes 1.4. Eventos 2. Etiquetas 3. Botones 4. Campos de Texto



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			<ul style="list-style-type: none"> 5. Áreas de Texto 6. Listas combinadas 7. CheckBox 8. RadioButton 9. Scroll bars 10. Menús y Submenús 11. Ventanas emergentes 12. EVENTOS <ul style="list-style-type: none"> 12.1. Eventos de botón 12.2. Eventos de listas combinadas 12.3. Eventos de checkbox y radiobutton 12.4. Eventos de menús 12.5. Eventos de ratón 12.6. Eventos de teclado 12.7. Eventos de foco FUNDAMENTOS DE BASES DE DATOS <ul style="list-style-type: none"> 1. Estructura de una base de datos 2. Creación de una base de datos 3. Lenguaje de consulta de datos SQL (Structured Query Language) 4. Consultas, actualizaciones, modificaciones, borrado de registros HILOS <ul style="list-style-type: none"> 1. Conceptos básicos sobre hilos 2. Ejecución en segundo plano 3. Creación de un hilo 4. Estado y control de un hilo 5. Sincronización de hilos



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			PROGRAMACIÓN PARA REDES 1. Sockets 2. Aplicaciones Cliente-Servidor 3. Clientes múltiples
AE7	Aportar soluciones creativas a problemas de ingeniería mecatrónica de manera autónoma y en equipo.	- Diseñar y trabajar eficientemente en equipos de desarrollo de software aplicando las metodologías de desarrollo de software y programación orientada a objetos para lograr soluciones eficientes y robustas. - Aplicar relaciones interpersonales, profesionales y colaborativas en el desarrollo de soluciones de software.	REPASO DE PROGRAMACIÓN ORIENTADA A OBJETOS 1. Creación de Clases y Objetos 2. Alcance de las clases (Scope) 3. Métodos 4. Composición 5. Ámbito de las variables 6. Paquetes de clases 7. Modificadores de acceso (default, public, private, protected) 8. Herencia 9. Polimorfismo 10. Clases abstractas 11. Qué es una excepción 12. Manejo de excepciones (try/catch/finally) 13. Métodos que lanzan excepciones explícitamente DISEÑO DE INTERFAZ GRÁFICA 1. Definiciones: 1.1. Frame 1.2. Contenedores 1.3. Componentes 1.4. Eventos 2. Etiquetas 3. Botones



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			<p>4. Campos de Texto</p> <p>5. Áreas de Texto</p> <p>6. Listas combinadas</p> <p>7. CheckBox</p> <p>8. RadioButton</p> <p>9. Scroll bars</p> <p>10. Menús y Submenús</p> <p>11. Ventanas emergentes</p> <p>12. EVENTOS</p> <p>12.1. Eventos de botón</p> <p>12.2. Eventos de listas combinadas</p> <p>12.3. Eventos de checkbox y radiobutton</p> <p>12.4. Eventos de menús</p> <p>12.5. Eventos de ratón</p> <p>12.6. Eventos de teclado</p> <p>12.7. Eventos de foco</p> <p>FUNDAMENTOS DE BASES DE DATOS</p> <p>1. Estructura de una base de datos</p> <p>2. Creación de una base de datos</p> <p>3. Lenguaje de consulta de datos SQL (Structured Query Language)</p> <p>4. Consultas, actualizaciones, modificaciones, borrado de registros</p> <p>HILOS</p> <p>1. Conceptos básicos sobre hilos</p> <p>2. Ejecución en segundo plano</p> <p>3. Creación de un hilo</p> <p>4. Estado y control de un hilo</p>



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			5. Sincronización de hilos PROGRAMACIÓN PARA REDES 1. Sockets 2. Aplicaciones Cliente-Servidor 3. Clientes múltiples

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Diseñar aplicaciones que se ejecuten en Ambientes Visuales de Usuario (GUI Graphical User Interface). Así como utilizar los recursos del Sistema Operativo para aplicarlos a situaciones de la vida real, en competencias industriales como lo son las comunicaciones, aplicaciones en tiempo real, así como la construcción de aplicaciones científicas y de investigación en las áreas de Ingeniería.		
Atributos (competencia específica) de la asignatura		
Conocer la Arquitectura de funcionamiento de los Sistemas Operativos de 32 y 64 bits; aplicar de manera efectiva la Programación Orientada a Objetos en el Análisis, Diseño e Implementación de Sistemas de Software Competentes en el Mercado, conocer las herramientas y técnicas de diseño de interfaces gráficas de usuario con el objetivo de que sus soluciones posean interacción eficaz con el usuario. Así como integrar el conocimiento adquirido de materias adyacentes y su experiencia para resolver problemas utilizando la computación y formar una sólida base de conocimientos en programación para aplicarlos a su área profesional.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
<ul style="list-style-type: none"> - Identificar soluciones informáticas con interfaz de usuario y bases de datos aplicables a ambientes cliente-servidor. - Conocer la Arquitectura de funcionamiento de los Sistemas Operativos de 32 y 64 bits. - Formar una sólida base de conocimientos en programación. 	<ul style="list-style-type: none"> - Utilizar tecnología de Desarrollo de Software tanto en entornos independientes de desarrollo (stand alone) como en ambientes integrados de Desarrollo (IDE). - Implementar soluciones informáticas con interfaz de usuario, bases de datos aplicables a ambientes cliente-servidor. 	Integrar equipos de trabajo multidisciplinario que involucre los ámbitos científicos, tecnológicos, socioculturales y humanísticos.
Producto integrador de la asignatura, considerando los avances por unidad		
Proyecto que incluya el desarrollo de una solución informática aplicable a la industria, comercio, negocios, educativos o sociales que resuelva una necesidad específica.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "REPASO DE PROGRAMACIÓN ORIENTADA A OBJETOS"

Número y nombre de la unidad: 1. REPASO DE PROGRAMACIÓN ORIENTADA A OBJETOS							
Tiempo y porcentaje para esta unidad:		Teoría:	8 horas	Práctica:	12 horas	Porcentaje del programa:	15.56%
Aprendizajes esperados:		Desarrollar soluciones a necesidades específicas utilizando programación orientada a objetos a nivel con sola					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Creación de Clases y Objetos 2. Alcance de las clases (Scope) 3. Métodos 4. Composición 5. Ámbito de las variables 6. Paquetes de clases 7. Modificadores de acceso (default, public, private, protected) 8. Herencia 9. Polimorfismo 10. Clases abstractas	<p>Saber:</p> <ul style="list-style-type: none"> - Identificar las bases de la Programación Orientada a Objetos. <p>Saber hacer:</p> <ul style="list-style-type: none"> - Desarrollar aplicaciones prácticas de la vida cotidiana utilizando el paradigma "Orientado a Objetos" <p>Ser:</p> <ul style="list-style-type: none"> - Se desenvuelve con valores éticos profesionales y disciplina. - Participa en ambientes colaborativos. - Lidera y participa activamente en las actividades de equipo. 	<p>Estrategia Pre-instruccionales</p> <ul style="list-style-type: none"> - Exposición del docente con ayuda de herramientas didácticas electrónicas. <p>Estrategia Co-instruccionales</p> <ul style="list-style-type: none"> - Identificación de datos respecto a los contenidos propuestos en la unidad. - Uso de herramientas electrónicas para apoyo didáctico. - Elaboración de mapas mentales y/o conceptuales. - Resolución de dinámicas, tareas, trabajos y/o actividades. <p>Estrategia Post-instruccionales</p> <ul style="list-style-type: none"> - Uso de software para simulación por computadoras de las redes o circuitos, aprendidos en la unidad. 	<p>Evaluación diagnóstica</p> <ul style="list-style-type: none"> - Autoevaluación de Programación Orientada a Objetos. <p>Evaluación sumativa</p> <ul style="list-style-type: none"> - Examen parcial teórico. - Examen práctico. <p>Evaluación formativa</p> <ul style="list-style-type: none"> - Prácticas de laboratorio. - Rúbrica de las prácticas. - Lista de cotejo. 	1er avance de proyecto que incluye las prácticas y conocimientos de Programación Orientada a Objetos			
Bibliografía							
Deitel, H. M. & Deitel, P. J. (2008). Como programar en Java. México: Prentice Hall.							
Dean, J.S.; Dean, R.H. (2009). Introducción a la Programación con Java. México: Mc Graw Hill.							

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Repaso de programación orientada a objetos."

Número y nombre de la unidad:		1. Repaso de programación orientada a objetos.					
Tiempo y porcentaje para esta unidad:		Teoría:	6 horas	Práctica:	8 horas	Porcentaje del programa:	15.56%
Aprendizajes esperados:		Desarrollar soluciones a necesidades específicas utilizando programación orientada a objetos a nivel consola.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Creación de Clases y Objetos. 2. Alcance de las clases (Scope). 3. Métodos. 4. Composición. 5. Ámbito de las variables. 6. Paquetes de clases. 7. Modificadores de acceso (default, public, private, protected). 8. Herencia. 9. Polimorfismo. 10. Clases abstractas.	Saber: - Identificar las bases de la Programación Orientada a Objetos. Saber hacer: - Desarrollar aplicaciones prácticas de la vida cotidiana utilizando el paradigma "Orientado a Objetos" Ser: - Se desenvuelve con valores éticos profesionales y disciplina. - Participa en ambientes colaborativos. - Lidera y participa activamente en las actividades de equipo.	Estrategia Pre-instruccionales: - Identificar conocimientos previos. Estrategia Co-instruccionales: - Exposición del docente con ayuda de herramientas didácticas electrónicas. - Identificación de datos respecto a los contenidos propuestos en la unidad. - Uso de herramientas electrónicas para apoyo didáctico. - Elaboración de mapas mentales y/o conceptuales. - Resolución de dinámicas, tareas, trabajos y/o actividades. Estrategia Post-instruccionales: - Uso de software para simulación por computadoras de las redes o circuitos,	Evaluación diagnóstica: - Autoevaluación de Programación. Orientada a Objetos. Evaluación formativa: - Prácticas de laboratorio. - Actividades, tareas. Instrumentos de evaluación: - Rúbrica de las prácticas. - Lista de cotejo. Evaluación sumativa: - Examen parcial teórico. - Examen práctico.	Primer avance de proyecto que incluye las prácticas y conocimientos de Programación Orientada a Objetos.			



Continuación: Tabla 4.2. Desglose específico de la unidad "Repaso de programación orientada a objetos."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
		aprendidos en la unidad.		
Bibliografía				
<p>- Deitel, H. M.; Deitel, P. J. (2008). Como programar en Java. México: Prentice Hall.</p> <p>- Dean, J. S.; Dean, R. H. (2009). Introducción a la Programación con Java. México: Mc Graw Hill.</p>				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Manejo de excepciones."

Número y nombre de la unidad: 2. Manejo de excepciones.							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	6 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados: Preparar sus desarrollos en software para la ocurrencia de posibles excepciones (errores) durante la ejecución de los mismos.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Qué es una excepción. 2. Manejo de excepciones (try/catch/finally). 3. Métodos que lanzan excepciones explícitamente.	<p>Saber:</p> <ul style="list-style-type: none"> - Identificar las circunstancias de excepción en el flujo de sus programas. <p>Saber hacer:</p> <ul style="list-style-type: none"> - Agregar a sus desarrollos de software la capacidad de reaccionar a situaciones de error. <p>Ser:</p> <ul style="list-style-type: none"> - Se desenvuelve con valores éticos profesionales y disciplina. - Participa en ambientes colaborativos. - Lidera y participa activamente en las actividades de equipo. 	<p>Estrategia Pre-instruccionales</p> <ul style="list-style-type: none"> - Exposición del docente con ayuda de herramientas didácticas electrónicas. <p>Estrategia Co-instruccionales</p> <ul style="list-style-type: none"> - Identificación de datos respecto a los contenidos propuestos en la unidad. - Uso de herramientas electrónicas para apoyo didáctico. - Elaboración de mapas mentales y/o conceptuales. - Resolución de dinámicas, tareas, trabajos y/o actividades. <p>Estrategia Post-instruccionales</p> <ul style="list-style-type: none"> - Uso de software para simulación por computadoras de las redes o circuitos, aprendidos en la unidad. 	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> - Prácticas de laboratorio. - Actividades, tareas. <p>Instrumentos de evaluación:</p> <ul style="list-style-type: none"> - Rúbrica de las prácticas. - Lista de cotejo. <p>Evaluación sumativa:</p> <ul style="list-style-type: none"> Examen parcial teórico. Examen práctico. 	<p>Primer avance de proyecto que incluye las prácticas y conocimientos de Programación Orientada a Objetos.</p>			



Continuación: Tabla 4.3. Desglose específico de la unidad "Manejo de excepciones."

Bibliografía

- Deitel, H. M.; Deitel, P. J. (2008). Como programar en Java. México: Prentice Hall.
- Dean, J. S.; Dean, R. H. (2009). Introducción a la Programación con Java. México: Mc Graw Hill.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad "Diseño de interfaz gráfica."

Número y nombre de la unidad: 3. Diseño de interfaz gráfica.							
Tiempo y porcentaje para esta unidad:		Teoría:	6 horas	Práctica:	10 horas	Porcentaje del programa:	17.78%
Aprendizajes esperados: Desarrollar interfaz gráfica (GUI) para interacción con el usuario final.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Definiciones. 1.1. Frame. 1.2. Contenedores. 1.3. Componentes. 1.4. Eventos. 2. Etiquetas. 3. Botones. 4. Campos de Texto. 5. Áreas de Texto. 6. Listas combinadas. 7. CheckBox. 8. RadioButton. 9. Scroll bars. 10. Menús y Submenús. 11. Ventanas emergentes. 12. EVENTOS. 12.1. Eventos de botón. 12.2. Eventos de listas combinadas. 12.3. Eventos de checkbox y radiobutton. 12.4. Eventos de menús.	Saber: - Identificar los componentes de las interfaces gráficas y su interacción con el usuario final de la aplicación. Saber hacer: - Utilizar los diferentes componentes de las interfaces gráficas Ser: - Se desenvuelve con valores éticos profesionales y disciplina. - Participa en ambientes colaborativos. - Lidera y participa activamente en las actividades de equipo.	Estrategia Pre-instruccionales - Exposición del docente con ayuda de herramientas didácticas electrónicas. Estrategia Co-instruccionales - Identificación de datos respecto a los contenidos propuestos en la unidad. - Uso de herramientas electrónicas para apoyo didáctico. - Elaboración de mapas mentales y/o conceptuales. - Resolución de dinámicas, tareas, trabajos y/o actividades. Estrategia Post-instruccionales - Uso de software para simulación por computadoras de las redes o circuitos,	Evaluación formativa - Prácticas de laboratorio. - Actividades, tareas. Instrumento de evaluación: - Rúbrica de las prácticas. - Lista de cotejo. Evaluación sumativa - Examen parcial teórico. - Examen práctico.	- 2do avance de proyecto que incluye las prácticas y conocimientos del diseño de interfaz gráfica2.			



Continuación: Tabla 4.4. Desglose específico de la unidad "Diseño de interfaz gráfica."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
12.5. Eventos de ratón. 12.6. Eventos de teclado. 12.7. Eventos de foco.		aprendidos en la unidad.		
Bibliografía				
- Deitel, H. M.; Deitel, P. J. (2008). Como programar en Java. México: Prentice Hall. - Dean, J. S.; Dean, R. H. (2009). Introducción a la Programación con Java. México: Mc Graw Hill.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.5. Desglose específico de la unidad "Fundamentos de bases de datos."

Número y nombre de la unidad: 4. Fundamentos de bases de datos.							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	6 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados: Implementar Bases de Datos para el manejo seguro de información en los Desarrollos de Software.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Estructura de una base de datos. 2. Creación de una base de datos. 3. Lenguaje de consulta de datos SQL (Structured Query Language). 4. Consultas, actualizaciones, modificaciones, borrado de registros.	Saber: - Identificar los fundamentos de las bases de de datos. Saber hacer: - Implementa Bases de Datos en las soluciones con interfaz gráfica. - Aplicar los fundamentos de las bases de datos. Ser: - Se desenvuelve con valores éticos profesionales y disciplina. - Participa en ambientes colaborativos. - Lidera y participa activamente en las actividades de equipo.	Estrategia Pre-instruccionales - Exposición del docente con ayuda de herramientas didácticas electrónicas. Estrategia Co-instruccionales - Identificación de datos respecto a los contenidos propuestos en la unidad. - Uso de herramientas electrónicas para apoyo didáctico. - Elaboración de mapas mentales y/o conceptuales. - Resolución de dinámicas, tareas, trabajos y/o actividades. Estrategia Post-instruccionales - Uso de software para simulación por computadoras de las redes o circuitos, aprendidos en la unidad.	Evaluación formativa: - Prácticas de laboratorio. - Actividades, tareas. Instrumentos de evaluación: - Rúbrica de las prácticas. - Lista de cotejo. Evaluación sumativa: - Examen parcial teórico. - Examen práctico.	- 2do avance de proyecto que incluye las prácticas y conocimientos del diseño de interfaz gráfica.			



Continuación: Tabla 4.5. Desglose específico de la unidad "Fundamentos de bases de datos."

Bibliografía

- Deitel, H. M.; Deitel, P. J. (2008). Como programar en Java. México: Prentice Hall.
- Dean, J. S.; Dean, R. H. (2009). Introducción a la Programación con Java. México: Mc Graw Hill.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.6. Desglose específico de la unidad "Hilos."

Número y nombre de la unidad: 5. Hilos.							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	6 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados: Implementar procesos en segundo plano mediante ?Hilos? (Threads) en los desarrollos de software.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Conceptos básicos sobre hilos. 2. Ejecución en segundo plano. 3. Creación de un hilo. 4. Estado y control de un hilo. 5. Sincronización de hilos.	Saber: - Identificar las características de los procesos en segundo plano mediante el uso de "hilos" (threads). Saber hacer: - Agregar a sus desarrollos de software procesos en segundo plano para optimizar el desempeño de los mismos. Ser: - Se desenvuelve con valores éticos profesionales y disciplina. - Participa en ambientes colaborativos. - Lidera y participa activamente en las	Estrategia Pre-instruccionales - Exposición del docente con ayuda de herramientas didácticas electrónicas. Estrategia Co-instruccionales - Identificación de datos respecto a los contenidos propuestos en la unidad. - Uso de herramientas electrónicas para apoyo didáctico. - Elaboración de mapas mentales y/o conceptuales. - Resolución de dinámicas, tareas, trabajos y/o actividades. Estrategia Post-instruccionales - Uso de software para simulación por computadoras de las redes o circuitos, aprendidos en la unidad.	Evaluación formativa: - Prácticas de laboratorio. - Actividades, tareas. Instrumentos de evaluación: - Rúbrica de las prácticas. - Lista de cotejo. Evaluación sumativa: - Examen parcial teórico. - Examen práctico.	- Entrega final de un proyecto global aplicando todo lo visto en el curso.			



Continuación: Tabla 4.6. Desglose específico de la unidad "Hilos."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	actividades de equipo.			
Bibliografía				
- Deitel, H. M.; Deitel, P. J. (2008). Como programar en Java. México: Prentice Hall. - Dean, J. S.; Dean, R. H. (2009). Introducción a la Programación con Java. México: Mc Graw Hill.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.7. Desglose específico de la unidad "Programación para redes."

Número y nombre de la unidad: 6. Programación para redes.							
Tiempo y porcentaje para esta unidad:		Teoría:	4 horas	Práctica:	6 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados: Desarrollar aplicaciones Cliente-Servidor mediante el uso de sockets y otros mecanismos de comunicación entre equipos de cómputo.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Sockets. 2. Aplicaciones Cliente-Servidor. 3. Clientes múltiples.	<p>Saber:</p> <ul style="list-style-type: none"> - Identificar las circunstancias de las comunicaciones vía "sockets" u otros protocolos para conectar varios equipos de cómputo e intercambiar información. - Comprender el paradigma "cliente-servidor" <p>Saber hacer:</p> <ul style="list-style-type: none"> - Agregar a sus desarrollos de software el uso de esquema "cliente-servidor" para intercambio y uso de información. <p>Ser:</p> <ul style="list-style-type: none"> - Se desenvuelve con valores éticos profesionales y disciplina. 	<p>Estrategia Pre-instruccionales</p> <ul style="list-style-type: none"> - Exposición del docente con ayuda de herramientas didácticas electrónicas. <p>Estrategia Co-instruccionales</p> <ul style="list-style-type: none"> - Identificación de datos respecto a los contenidos propuestos en la unidad. - Uso de herramientas electrónicas para apoyo didáctico. - Elaboración de mapas mentales y/o conceptuales. - Resolución de dinámicas, tareas, trabajos y/o actividades. <p>Estrategia Post-instruccionales</p> <ul style="list-style-type: none"> - Uso de software para simulación por computadoras de las redes o circuitos, aprendidos en la unidad. 	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> - Prácticas de laboratorio. - Actividades, tareas. <p>Instrumentos de evaluación:</p> <ul style="list-style-type: none"> - Rúbrica de las prácticas. - Lista de cotejo. <p>Evaluación sumativa:</p> <ul style="list-style-type: none"> - Examen parcial teórico. - Examen práctico. 	<ul style="list-style-type: none"> - Entrega final de un proyecto global aplicando todo lo visto en el curso. 			



Continuación: Tabla 4.7. Desglose específico de la unidad "Programación para redes."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<ul style="list-style-type: none">- Participa en ambientes colaborativos.- Lidera y participa activamente en las actividades de equipo.			
Bibliografía				
<ul style="list-style-type: none">- Deitel, H. M.; Deitel, P. J. (2008). Como programar en Java. México: Prentice Hall.- Dean, J. S.; Dean, R. H. (2009). Introducción a la Programación con Java. México: Mc Graw Hill.				



V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
<p>Carrera(s): - Ingeniería en Computación.</p> <p>- Ingeniería en Desarrollo de Software. o carrera afín</p> <ul style="list-style-type: none">- Experiencia en los campos de la Ingeniería en Computación y Desarrollo de Software.- Experiencia mínima de dos años- Ingeniero