

**PROGRAMA DE ESTUDIO POR COMPETENCIAS  
LENGUAJE DE PROGRAMACIÓN ORIENTADA A OBJETOS**

**I. IDENTIFICACIÓN DEL CURSO**

<b>ESPACIO EDUCATIVO:</b> Facultad de Ingeniería						
<b>LICENCIATURA:</b> Ingeniería de Computación				<b>ÁREA DE DOCENCIA:</b> Programación de Ingeniería de Software		
<b>AÑO DE APROBACIÓN POR EL CONSEJO UNIVERSITARIO:</b>						
<b>APROBACIÓN POR LOS HH. CONSEJOS ACADÉMICO Y DE GOBIERNO</b>		<b>FECHA:</b>		<b>PROGRAMA ELABORADO POR:</b> Ing. Victor Sarabia Ortíz		<b>PROGRAMA REVISADO POR:</b> Integrantes de la Academia de Programación de Ingeniería de Software
				<b>FECHA DE ELABORACIÓN :</b> Junio 2011		<b>FECHA DE REVISIÓN :</b> Junio 2011
<b>CLAVE</b>	<b>HORAS DE TEORÍA</b>	<b>HORAS DE PRÁCTICA</b>	<b>TOTAL DE HORAS</b>	<b>CRÉDITOS</b>	<b>TIPO DE CURSO</b>	<b>NÚCLEO DE FORMACIÓN</b>
L41092	2	1	3	5	Curso-Laboratorio	Integral
<b>PRERREQUISITOS</b> Programación Orientada a Objetos		<b>UNIDAD DE APRENDIZAJE ANTECEDENTE</b> Ninguna		<b>UNIDAD DE APRENDIZAJE CONSECUENTE</b> Ninguna		
<b>PROGRAMAS EDUCATIVOS O ESPACIOS ACADÉMICOS EN LOS QUE SE IMPARTE:</b> Licenciatura en Ingeniería en Computación (Facultad. de Ingeniería, Centros Universitarios: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco, Valle de México, Valle de Teotihuacán, Zumpango)						

## II. PRESENTACIÓN DEL PROGRAMA

La “Programación Orientada a Objetos”, desde principios de los años noventa, es el paradigma predominante en el desarrollo de sistemas de software. En el mercado se encuentran un sin número de aplicaciones que prueban que este paradigma es un excelente medio para lograr dos de las metas más importantes en el desarrollo de sistemas de software: robustez y eficiencia. La larga vigencia de esta tecnología ha propiciado en la actualidad cuente con múltiples recursos, entre los que podemos mencionar: lenguajes de programación, metodologías y patrones de diseño así como herramientas de desarrollo (plataformas, IDEs, etc.).

Para poder utilizar de forma efectiva este paradigma de programación es necesario además del dominio de el paradigma en si mismo, tener criterios para poder discernir respecto al universo de herramientas y demás recursos que existen para la creación de sistemas. Para cada proyecto de software los profesionales de esta área deben poseer conocimientos que les permitan seleccionar de forma adecuada los recursos a utilizar; para que el sistema además de cumplir con los requisitos (tanto funcionales como no funcionales) del cliente, pueda ser desarrollado en el menor tiempo y con el menor esfuerzo posibles.

La unidad de aprendizaje “Lenguaje de programación orientado a objetos” permitirá al discente tomar un perfil más profesional respecto a la disciplina de desarrollo de software. Ampliando su visión respecto al paradigma de orientación a objetos mediante las aplicaciones de patrones de diseño aplicados directamente a Frameworks de desarrollo, así como en la API de un lenguaje de programación orientado a objetos. Así como mediante el uso de herramientas y plataformas profesionales para el desarrollo de sistemas.

## III. LINEAMIENTOS DE LA UNIDAD DE APRENDIZAJE

DEL DOCENTE	DEL DISCENTE
<ul style="list-style-type: none"><li>• Cubrir el programa en su totalidad</li><li>• Asistencia al 100% de las sesiones</li><li>• Dar a conocer oportunamente los resultados de exámenes y tareas</li></ul>	<ul style="list-style-type: none"><li>• Asistencia de por lo menos 80%</li><li>• Puntualidad</li><li>• Respeto al profesor y a los compañeros de clase</li></ul>

#### **IV. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE**

Familiarizar al docente con un lenguaje de programación orientado a objetos, los patrones de diseño en las que se sustentan algunas de las clases en interfaces de su API, así como con una plataforma de desarrollo, que le permita la aplicación de ésta para el desarrollo de software de forma eficiente y efectiva.

#### **V. COMPETENCIAS GENÉRICAS**

- Seleccionar la plataforma de desarrollo, patrones de diseño y herramientas de desarrollo para crear sistemas de software.
- Utilizar las herramientas seleccionadas para obtener un sistema de software que satisfaga los requerimientos de un cliente.

#### **VI. ÁMBITOS DE DESEMPEÑO PROFESIONAL**

Desarrollo de software  
Arquitectura de software  
Ingeniería de software  
Selección de herramientas de desarrollo

#### **VII. ESCENARIOS DE APRENDIZAJE**

Aula y sala de cómputo

#### **VIII. ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE**

1. Plataforma y lenguaje ( 2 semanas)
2. Clases, objetos y lectura de datos del teclado ( 2 semanas)
3. Enumeraciones ( 1 semanas)

4. Clases Abstracta e Interfaces ( 2 semanas)
5. Colecciones e iteradores ( 1 semanas)
6. Clases genéricas ( 1 semanas)
7. Uis ( 2 semanas)
8. Persistencia ( 2 semanas)
9. Tópicos Adicionales ( 3 semanas)
  - Threads
  - Networking.

#### IX. DESARROLLO DE LA UNIDAD DE APRENDIZAJE

UNIDAD DE COMPETENCIA I:	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES / VALORES
Plataforma y lenguaje	Clasificaciones Características de los lenguaje de Programación Orientada a Objetos Características de la plataforma Instalación y preparación del lenguaje y el IDE	Instalación del lenguaje preparación de variables de entorno configuración del IDE	Espíritu crítico Trabajo en equipo Respeto y tolerancia Honestidad
Estrategias didácticas: Exposición por parte del profesor Lectura por parte de los estudiantes Discusión por equipos para contrastar las características de distintos LPOO Plenaria para homologar conclusiones		<b>RECURSOS REQUERIDOS:</b>  Pizarrón Artículos sobre características de distintos lenguajes y tecnologías.	<b>TIEMPO DESTINADO:</b>  6 horas
<b>CRITERIOS DE DESEMPEÑO</b>	<b>EVIDENCIAS</b>		
		<b>DESEMPEÑO</b>	<b>PRODUCTOS</b>

Participación activa Trabajo en equipo	Trabajo en equipo	Matriz de comparación de características de LPOO
---	-------------------	--

UNIDAD DE COMPETENCIA II:	ELEMENTOS DE COMPETENCIA		
	Conocimientos	Habilidades	Actitudes / Valores
Clases, objetos y lectura de datos del teclado	Criterios para definición de clases Patrón alta cohesión Delegación de responsabilidades a una clase Herencia vs agregación Conversiones de tipos el operador instanceOf	Diseño y codificación, de clases aplicando el patrón alta cohesión. Adquirir datos desde el teclado mediante Scanner, JoptionPane o BufferdReader	Espíritu crítico Trabajo en equipo Respeto y tolerancia Honestidad Superación personal
<b>ESTRATEGIAS DIDÁCTICAS:</b> Exposición por parte del profesor Lectura por parte de los estudiantes Programación por parte de los dicentes. Plenaria para homologar conclusiones		<b>RECURSOS REQUERIDOS:</b> Pizarrón Sala de computo JDSK 6 o posterior NetBeans / JDeveloper Videoprojector	<b>TIEMPO DESTINADO:</b> 6 horas
CRITERIOS DE DESEMPEÑO	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	
Apego a estándares de nomenclatura Buen estilo de programación Uso de visibilidad apropiada en clases, atributos y métodos Pruebas unitarias apropiadas para cada clase y método	Clases desarrolladas de forma modular, cohesivas y débilmente acopladas	Clase Pruebas unitarias	

UNIDAD DE COMPETENCIA III	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES / VALORES
Enumeraciones	Tipos de clases Aspectos de diseño aplicables para enumeraciones Características de una enumeración Objetos de enumeraciones Métodos y constructores en enumeraciones. Uso de enumeraciones	Diseño y codificación, enumeraciones Uso de enumeraciones	Espíritu crítico Trabajo en equipo Respeto y tolerancia Honestidad Superación personal
<b>ESTRATEGIAS DIDÁCTICAS</b> Exposición por parte del profesor Lectura por parte de los estudiantes Programación por parte de los docentes. Plenaria para homologar conclusiones		<b>RECURSOS REQUERIDOS</b> Pizarrón Sala de computo JDK 6 o posterior NetBeans / JDeveloper Videoprojector	<b>TIEMPO DESTINADO</b> 3 horas
<b>CRITERIOS DE DESEMPEÑO</b>		<b>EVIDENCIAS</b>	
		<b>DESEMPEÑO</b>	<b>PRODUCTOS</b>
Apego a estándares de nomenclatura Buen estilo de programación Enumeraciones con atributos, métodos y constructoras		Enumeraciones desarrolladas Clase de uso de enumeraciones	Clase de enumeración Clase que usa enumeraciones

UNIDAD DE COMPETENCIA IV	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES / VALORES
Clases Abstracta e Interfaces	Aspectos de diseño relacionados con el uso y desarrollo de interfaces	Diseño y codificación, interfaces	Espíritu crítico Trabajo en equipo

	<p>           Criterios para la definición de interfaces y clases abstractas            Patrón bajo acoplamiento            Las Interfaces y la Delegación de responsabilidades a una clase            Herencia en Interfaces            Implementación de Interfaces            Forzamiento de tipos mediante interfaces            Algunas interfaces importantes: Serializable, comparable, etc.         </p>	<p>           Adquirir datos desde el teclado mediante Scanner, JOptionPane o BufferedReader         </p>	<p>           Respeto y tolerancia            Honestidad            Superación personal         </p>
<p> <b>ESTRATEGIAS DIDÁCTICAS:</b>            Exposición por parte del profesor            Lectura por parte de los estudiantes            Programación por parte de los docentes.            Plenaria para homologar conclusiones         </p>	<p> <b>RECURSOS REQUERIDOS:</b>            Pizarrón            Sala de computo            JDK 6 o posterior            NetBeans            Videoprojector         </p>	<p> <b>TIEMPO DESTINADO:</b>            6 horas         </p>	
		<b>EVIDENCIAS</b>	
	<b>CRITERIOS DE DESEMPEÑO</b>	<b>DESEMPEÑO</b>	<b>PRODUCTOS</b>
	<p>           Apego a estándares de nomenclatura            Buen estilo de programación            Apropia implementación y uso de colecciones e iteradores         </p>	<p>           Desarrollo de clases de forma modular, cohesivas y débilmente acopladas         </p>	<p>           Clase            Pruebas unitarias         </p>

<b>UNIDAD DE COMPETENCIA V</b>	<b>ELEMENTOS DE COMPETENCIA</b>		
	<b>CONOCIMIENTOS</b>	<b>HABILIDADES</b>	<b>ACTITUDES / VALORES</b>
Colecciones e iteradores	<p>           Características de las colecciones.            Colecciones de longitud fija y de longitud variable            Operaciones básicas de una         </p>	<p>           Discernir entre el tipo de colección más conveniente a un problema            Uso de Colecciones         </p>	<p>           Espíritu crítico            Trabajo en equipo            Respeto y tolerancia            Honestidad         </p>

	colecciones Uso de iteradores sobre colecciones Definición y uso de colecciones Algunos tipos de colecciones como Vector, List, LinkedList, ArrayList, etc.	Implementación de colecciones	Superación personal
<b>ESTRATEGIAS DIDÁCTICAS:</b> Exposición por parte del profesor Lectura por parte de los estudiantes Programación por parte de los docentes. Plenaria para homologar conclusiones		<b>RECURSOS REQUERIDOS:</b> Pizarrón Sala de computo JDSK 6 o posterior NetBeans / JDeveloper cañón	<b>TIEMPO DESTINADO:</b> 1 horas
<b>CRITERIOS DE DESEMPEÑO</b>		<b>EVIDENCIAS</b>	
		<b>DESEMPEÑO</b>	<b>PRODUCTOS</b>
Apego a estándares de nomenclatura Buen estilo de programación Uso apropiada de colecciones e iteradores Pruebas unitarias apropiadas		Desarrollo de colecciones e iteradores. Uso apropiado de colecciones e iteradores	Clase de colección Iterador par la colección Uso de de colección e iterador Pruebas unitarias

UNIDAD DE COMPETENCIA VI	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES / VALORES
Clases genéricas	Uso de clases genéricas Restricciones de tipos en clases genéricas	Utilización de objetos <i>generics</i> : declaración instanciación y utilización	Espíritu crítico Trabajo en equipo Respeto y tolerancia

	Implementación de clases genéricas Herencia en clases genéricas	Crear clases <i>generics</i>	Honestidad Superación personal
<b>ESTRATEGIAS DIDÁCTICAS</b> Exposición por parte del profesor Lectura por parte de los estudiantes Programación por parte de los docentes. Plenaria para homologar conclusiones		<b>RECURSOS REQUERIDOS:</b> Pizarrón Sala de computo JDSK 6 o posterior NetBeans Videoprojector	<b>TIEMPO DESTINADO:</b> 3 horas
<b>CRITERIOS DE DESEMPEÑO</b>		<b>EVIDENCIAS</b>	
		<b>DESEMPEÑO</b>	<b>PRODUCTOS</b>
Apego a estándares de nomenclatura Buen estilo de programación Uso apropiado de objetos <i>generics</i> Pruebas unitarias apropiadas para cada clase y método		Desarrollo de clase que implemente un <i>generic</i> Desarrollo de clase que utilice un <i>generic</i>	Clase <i>generic</i> Clase con un <i>generic</i> Pruebas unitarias

UNIDAD DE COMPETENCIA VII:	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES / VALORES
Uis	Programación orientada a eventos Elementos de interfaz de usuario El patrón <i>observer</i> , la interfaz Observable Jerarquía de clases Desarrollo de UI con uso del IDE NetBeans Inner class		Espíritu crítico Trabajo en equipo Respeto y tolerancia Honestidad Superación personal

<b>ESTRATEGIAS DIDÁCTICAS:</b> Exposición por parte del profesor Lectura por parte de los estudiantes Programación por parte de los docentes. Plenaria para homologar conclusiones	<b>RECURSOS REQUERIDOS:</b> Pizarrón Sala de computo JDSK 6 o posterior NetBeans Videoprojector	<b>TIEMPO DESTINADO:</b> 6 horas
<b>CRITERIOS DE DESEMPEÑO</b>	<b>EVIDENCIAS</b>	
	<b>DESEMPEÑO</b>	<b>PRODUCTOS</b>
Apego a estándares de nomenclatura Buen estilo de programación Uso de visibilidad apropiada en clases, atributos y métodos Pruebas unitarias Interfaz de usuario con elementos y controles apropiados	Interfaz de usuario Controlador	Interfaz de usuario

<b>UNIDAD DE COMPETENCIA VIII:</b>	<b>ELEMENTOS DE COMPETENCIA</b>		
	<b>CONOCIMIENTOS</b>	<b>HABILIDADES</b>	<b>ACTITUDES / VALORES</b>
Persistencia	Paradigmas de persistencia en POO Patrón MVC Persistencia en archivos Persistencia de objetos serealizables XMLEncoder JDBC EJBs	Selección y codificación de diferentes formas de persistencia	Espíritu crítico Trabajo en equipo Respeto y tolerancia Honestidad Superación personal
<b>ESTRATEGIAS DIDÁCTICAS</b> Exposición por parte del profesor Lectura por parte de los estudiantes	<b>RECURSOS REQUERIDOS:</b> Pizarrón Sala de computo	<b>TIEMPO DESTINADO:</b> <b>6 horas</b>	

Desarrollo por parte de los dicentes. Plenaria para homologar conclusiones	JDSK 6 o posterior NetBeans Videoprojector	
<b>CRITERIOS DE DESEMPEÑO</b>	<b>EVIDENCIAS</b>	
	<b>DESEMPEÑO</b>	<b>PRODUCTOS</b>
Apego a estándares Buen estilo de programación Apego a MVC Pruebas unitarias apropiadas para cada clase y método	Desarrollo de objetos persistentes	Clase persistente Pruebas unitarias

<b>UNIDAD DE COMPETENCIA IX:</b>	<b>ELEMENTOS DE COMPETENCIA</b>		
	<b>CONOCIMIENTOS</b>	<b>HABILIDADES</b>	<b>ACTITUDES / VALORES</b>
Tópicos adicionales.	Threads Networking	Desarrollo de una aplicación con threads. Implementación de conectividad básica.	Espíritu crítico Trabajo en equipo Respeto y tolerancia Honestidad Superación personal
<b>ESTRATEGIAS DIDÁCTICAS</b> Exposición por parte del profesor Lectura por parte de los estudiantes Programación por parte de los dicentes. Plenaria para homologar conclusiones		<b>RECURSOS REQUERIDOS</b> Pizarrón Sala de computo JDSK 6 o posterior NetBeans Videoprojector	<b>TIEMPO DESTINADO:</b> 9 horas

CRITERIOS DE DESEMPEÑO	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
Apego a estándares Buen estilo de programación Uso de conceptos y clases correspondientes.	Desarrollo de aplicaciones que usen threads y conectividad	Aplicación

## X. EVALUACIÓN Y ACREDITACIÓN

### Evaluación

Para obtener la calificación se considerará desempeño regular durante el semestre:

- ❖ Tareas de investigación y prácticas 50 %
- ❖ Exámenes Parciales 50 %

Si el promedio obtenido mediante estos instrumentos es mayor o igual a 8.0 el alumno se considerará exento de presentar evaluación ordinaria.

### Evaluación ordinaria

Examen acumulativo	50 %
Proyecto final	50 %

### Evaluación extraordinaria

Examen acumulativo	50 %
Proyecto final	50 %

### Evaluación a título de suficiencia

Examen acumulativo	50 %
Proyecto final	50 %

### Acreditación:

1. Cumplir con el porcentaje de asistencia indicado en la legislación vigente.
2. Contar con una calificación aprobatoria tanto en la parte práctica (proyecto), como en la teórica (examen).

## XI. REFERENCIAS

### **BIBLIOGRAFÍA BÁSICA**

- ✓ Rod Jonson, J2EE Desing and Development, Worx
- ✓ Horstman C.Y, Cornell G., Java Core 2, volumen 1, séptima edición, Pearson – Prentice Hall,
- ✓ Mark Cade, Simmon Roberts, Sun Certified Enterprise Architect for J2EE Technology, Sun Press
- ✓ Jim Keogh, J2EE, Mac Graw Hill