

## 1. Datos Generales de la asignatura

<b>Nombre de la asignatura:</b>	Fundamentos de Programación
<b>Clave de la asignatura:</b>	AEF-1032
<b>SATCA<sup>1</sup>:</b>	3-2-5
<b>Carrera:</b>	Ingeniería Informática e Ingeniería en Tecnologías de la Información y Comunicaciones

## 2. Presentación

### Caracterización de la asignatura

La asignatura aporta al perfil del egresado, los conocimientos, habilidades, metodologías, así como capacidades de análisis y síntesis.

Es importante ya que permite plantear la solución de problemas susceptibles de ser computarizados, a través de técnicas y herramientas de modelado y codificación del paradigma orientado a objetos.

Esta asignatura identifica los fundamentos de la programación orientada a objetos, las metodologías y herramientas de programación a utilizar con este paradigma aplicando modelado de los elementos básicos (atributos y métodos).

Se relaciona con las asignaturas en donde se apliquen metodologías de programación.

### Intención didáctica

Este programa de estudios se sugiere eminentemente práctico, es decir, el docente propone el planteamiento de un problema y el estudiante deberá resolverlo mediante diversas técnicas, por mencionar algunas: herramientas de modelado y codificación; con el monitoreo del docente.

El tema uno, introduce al paradigma orientado a objetos, asegurando que comprenda los conceptos, permitiéndole reconocer contextos y sus objetos tangibles e intangibles que los conforman. Identifica diversos entornos de desarrollo que le permitan valorar ventajas y desventajas de cada uno.

En el segundo tema conoce el modelo de las 6'D y su aplicación en diversos ejemplos generando en el estudiante la capacidad de análisis para identificar problemas y plantear soluciones.

El tema tres, se enfoca en los conceptos de clases y objetos adquiriendo los conocimientos para modelar las propuestas sobre el contexto resaltando la jerarquía de clases y el instanciamiento de objetos.

En el tema cuatro se abordan las diferentes herramientas de programación orientada a objetos de manera formal, se analizan procesos simples que hacen uso de los comentarios, palabras reservadas, operadores, expresiones y tipos de datos, permitiendo adicionar funcionalidad a las clases mediante el uso de métodos. Promoviendo el uso de software de modelado para que se pruebe la generación automática de pseudocódigo en diferentes lenguajes, con la finalidad que identifique los elementos

<sup>1</sup> Sistema de Asignación y Transferencia de Créditos Académicos

lógicos comunes.

En el tema cinco se abordan las estructuras de control (entrada, salida, selección y repetición) y el estudiante las aplica en la programación de soluciones. Se resalta la importancia de que el docente profundice en la identificación de las estructuras selectivas y repetitivas ante diferentes contextos, lo cual permita establecer con claridad su aplicación de acuerdo a la problemática.

Es relevante que el docente a partir del tema 2, trabaje con los estudiantes en la identificación de un problema y promueva la aplicación gradual de las competencias adquiridas en cada tema, logrando integrar un proyecto de asignatura.

El docente deberá promover que el estudiante desarrolle las competencias genéricas para el análisis y resolución de problemas reales, así como las discusiones grupales y exposiciones que fomenten la competencia de expresión oral.

### 3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
<p>Instituto Tecnológico de Aguascalientes del 15 al 18 de junio de 2010.</p>	<p>Representantes de los Institutos Tecnológicos de: Centro Interdisciplinario de Investigación y Docencia en Educación Técnica, Acapulco, Aguascalientes, Apizaco, Boca Río, Celaya, Chetumal, Chihuahua, Chilpancingo, Chiná, Cd. Cuauhtémoc, Cd. Juárez, Cd. Madero, Cd. Victoria, Colima, Comitán, Cautla, Durango, El Llano de Aguascalientes, Huixquilucan, Valle Bravo, Guaymas, Huatabampo, Huejutla, Iguala, La Laguna, La Paz, La Zona Maya, León, Lerma, Linares, Los Mochis, Matamoros, Mazatlán, Mérida, Mexicali, Minatitlán, Nuevo Laredo, Orizaba, Pachuca, Puebla, Querétaro, Reynosa, Roque, Salina Cruz, Saltillo, San Luis Potosí, Tehuacán, Tepic, Tijuana, Tlaxiaco, Toluca, Torreón, Tuxtepec, Valle de Oaxaca, Veracruz, Villahermosa, Zacatecas,</p>	<p>Elaboración del programa de estudio equivalente en la Reunión Nacional de Implementación Curricular y Fortalecimiento Curricular de las asignaturas comunes por área de conocimiento para los planes de estudio actualizados del SNEST.</p>

	<p>Zacatepec, Altiplano de Tlaxcala, Coatzacoalcos, Cuautitlán Izcalli, Fresnillo, Irapuato, La Sierra Norte Puebla, Macuspana, Naranjos, Pátzcuaro, Poza Rica, Progreso, Puerto Vallarta, Tacámbaro, Tamazula Gordiano, Tlaxco, Venustiano Carranza, Zacapoaxtla, Zongólica y Oriente del Estado Hidalgo.</p>	
<p>Instituto Tecnológico de Morelia del 10 al 13 de septiembre de 2013.</p>	<p>Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Boca del Río, Celaya, CRODE Celaya, Cerro Azul, Chihuahua, Cd. Cuauhtémoc, Cd. Hidalgo, Cd. Juárez, Cd. Madero, Cd. Valles, Coacalco, Colima, Iguala, La Laguna, Lerdo, Los Cabos, Matamoros, Mérida, Morelia, Motúl, Múzquiz, Nuevo Laredo, Nuevo León, Oriente del Estado de México, Orizaba, Pachuca, Progreso, Purhepecha, Salvatierra, San Juan del Río, Santiago Papasquiari, Tantoyuca, Tepic, Tlatlauquitpec, Valle de Morelia, Venustiano Carranza, Veracruz, Villahermosa, Zacatecas y Zacatepec.</p>	<p>Reunión Nacional de Seguimiento Curricular de las Asignaturas Equivalentes del SNIT.</p>

#### 4. Competencia(s) a desarrollar

<b>Competencia(s) específica(s) de la asignatura</b>
<p>Aplica las herramientas de programación orientada a objetos, para modelar y desarrollar soluciones a diversos problemas del mundo real.</p>

#### 5. Competencias previas

<ul style="list-style-type: none"> <li>• Ninguna</li> </ul>
---

## 6. Temario

No.	Temas	Subtemas
1	Fundamentos de programación orientada a objetos	1.1. Evolución de la programación 1.2. Conceptos fundamentales de la programación orientada a objetos 1.3. Lenguajes orientados a objetos 1.4. Relaciones entre clases y objetos 1.5. Papel de clases y objetos en el análisis y el diseño 1.6. Entornos de programación
2	Metodología de solución de problemas	2.1 Descripción del problema 2.2 Definición de solución 2.3 Diseño de la solución 2.4 Desarrollo de la solución 2.5 Depuración y pruebas 2.6 Documentación 2.7 Solución de problemas aplicando la metodología
3	Acercamiento a las clases y objetos	3.1 Estructura de una clase 3.2 Elementos de una clase 3.3 Declaración de métodos 3.4 Métodos de clase y de instancia 3.6 Clase principal 3.7 Crear objetos
4	Herramientas de programación orientada a objetos	4.1 Simbología 4.2 Reglas para la construcción de diagramas 4.3 Pseudocódigo 4.4 Palabras reservadas y comentarios 4.5 Identificadores 4.5 Tipos de datos y conversiones 4.6 Operadores 4.7 Expresiones
5	Estructuras de control	5.1 Entrada y salida de datos 5.2 Selectivas 5.3 Repetitivas

## 7. Actividades de aprendizaje de los temas

Fundamentos de programación orientada a objetos	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Comprende los conceptos de la programación orientada a objetos, para aplicarlos en la resolución de problemas.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>Habilidades de gestión de información</li> </ul>	<ul style="list-style-type: none"> <li>Ilustrar en una línea de tiempo la evolución de la programación.</li> <li>Investigar conceptos fundamentales de la programación orientada a objetos.</li> <li>Identificar los objetos existentes (tangibles e intangibles) en un contexto dado.</li> <li>Describir los objetos mediante atributos.</li> </ul>

<p>(habilidad para buscar y analizar información proveniente de fuentes diversas).</p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Conocimientos básicos de la carrera.</li> <li>• Capacidad de comunicación oral y escrita.</li> </ul>	<ul style="list-style-type: none"> <li>• Seleccionar y reconocer un entorno de programación.</li> </ul>
Metodología de solución de problemas	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Conoce el modelo de las 6'D para la solución de problemas reales.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Capacidad para actuar en nuevas situaciones.</li> <li>• Habilidad para trabajar en forma autónoma.</li> <li>• Habilidades en el uso de las tecnologías de la información y de la comunicación.</li> </ul>	<ul style="list-style-type: none"> <li>• Investigar el modelo de las 6'D y elaborar un mapa conceptual.</li> <li>• Enlistar ejemplos de aplicación del modelo y exponerlo haciendo uso de las herramientas computacionales disponibles.</li> <li>• Elegir un problema del entorno, describirlo y proponer posibles soluciones.</li> </ul>
Acercamiento a las clases y objetos	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Conoce y aplica la estructura y formato de las clases y objetos para su modelado.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad creativa.</li> <li>• Capacidad para identificar, plantear y resolver problemas.</li> <li>• Capacidad de análisis y síntesis.</li> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Habilidades en el uso de las tecnologías de la información y de la comunicación.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar lenguajes de modelado orientado a objetos a través de una investigación y presentar un informe.</li> <li>• Identificar clases y objetos en un contexto y representarlos utilizando un lenguaje de modelado.</li> <li>• Modelar las clases y objetos del problema planteado en el tema anterior, utilizando alguna herramienta de software.</li> </ul>
Herramientas de programación orientada a objetos	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Conoce y aplica las herramientas de programación básicas para modelar y desarrollar la solución de problemas.</p>	<ul style="list-style-type: none"> <li>• Identificar las diferentes simbologías de modelado orientado a objetos, así como los diversos diagramas que existen a través de una investigación y presentar un informe.</li> <li>• Definir los conceptos de pseudocódigo,</li> </ul>

<p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Capacidad para actuar en nuevas situaciones.</li> <li>• Capacidad para identificar, plantear y resolver problemas.</li> <li>• Habilidades en el uso de las tecnologías de la información y de la comunicación.</li> </ul>	<p>palabras reservadas, comentarios, identificador, tipos de datos y sus conversiones, operadores, expresiones y a través de diversas fuentes de información y elaborar un mapa mental o conceptual.</p> <ul style="list-style-type: none"> <li>• Aplicar reglas para la construcción de diagramas (clases, transición de estados, casos de uso, entre otros) en diversos problemas planteados, utilizando herramientas de software.</li> <li>• Identificar y elaborar los diagramas aplicables al problema planteado en temas anteriores, utilizando alguna herramienta de software.</li> </ul>
Estructuras de control	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Conoce y aplica las estructuras de control para construir soluciones a problemas del entorno.</p> <p><b>Genéricas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad para identificar, plantear y resolver problemas.</li> <li>• Capacidad para formular y gestionar proyectos.</li> <li>• Capacidad de trabajo en equipo.</li> <li>• Habilidades en el uso de las tecnologías de la información y de la comunicación.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar las diferentes estructuras de entrada, salida, selectivas (simple, doble y múltiple) y repetitivas (fijas y variables) mediante una investigación y presentar un esquema gráfico que represente cada estructura.</li> <li>• Aplicar las estructuras de entrada y salida en problemas planteados.</li> <li>• Identificar las estructuras selectivas y/o repetitivas a utilizar acorde a las problemáticas establecidas.</li> <li>• Codificar la propuesta de solución al problema planteado en temas anteriores, utilizando un lenguaje orientado a objetos.</li> </ul>

**8. Práctica(s)**

<ul style="list-style-type: none"> <li>• Instalar un entorno de programación de un lenguaje orientado a objetos</li> <li>• Elaborar diagramas de clases e instanciamiento de objetos.</li> <li>• Elaborar diagramas de transición de estados, de casos de uso, de secuencia, entre otros.</li> <li>• Documentar la aplicación del modelo de las 6D's que incluya: redacción del problema. datos de entrada y salida, descripción de procesos, diseño de la solución.</li> <li>• Desarrollar aplicaciones que incluyan el manejo de operaciones básicas (aritméticas, lógicas, de asignación) y estructuras de control (entrada-salida, selectivas y repetitivas) para la resolución de problemas.</li> </ul>
--

**9. Proyecto de asignatura**

<ul style="list-style-type: none"> <li>• <b>Fundamentación:</b> marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.</li> </ul>
--

- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

### 10. Evaluación por competencias

Para evaluar las actividades de aprendizaje se recomienda solicitar: mapas conceptuales o mentales, reporte de investigación, reportes de prácticas, códigos de programas, estudio de casos, exposiciones en clase, problemarios, portafolio de evidencias, entre otros.

Para verificar el nivel del logro de las competencias del estudiante se recomienda utilizar: listas de cotejo, listas de verificación, matrices de valoración, guías de observación, rúbricas, entre otros.

### 11. Fuentes de información

1. Barry, P. Griffi D. (2009) *Head First Programming*. USA: O’Reilly.
2. Böck, H. (2012) *The Definitive Guide to NetBeans Platform 7*. USA:Apress
3. Ceballos, J. (2007) *Java 2 Lenguaje y aplicaciones*. España: Alfaomega.
4. Deitel, P. y Deitel, H. (2012) *Como programar en Java*. México:Prentice Hall.
5. Eckel, B. (2009) *Thinking in Java*. EUA: Prentice Hall.
6. Flores, J J. (2007) *Método de las 6’D UML – Pseudocódigo – Java Enfoque Algorítmico, Serie Textos Universitarios Facultad de Ingeniería y Arquitectura*. Perú: ed. Universidad de San Martín de Porres, (<http://books.google.com/>).
7. Friesen, J. (2012) *Beginning Java 7*. EUA:Apress.
8. Froufe, A. (2008) *Java 2 Manual de usuario y tutorial*. España: Alfaomega
9. Joyanes, L. y Zahonero, I. (2011) *Programación en Java 6*. España: McGraw Hill
10. Reese, R. (2012) *Oracle Certified Associate, Java SE 7 Programmer Study Guide*. :Packt Publishing. <http://www.amazon.com/Oracle-Certified-Associate-Programmer-Study/dp/1849687323>
11. Wu, T. (2008) *A comprehensive introduction to object-oriented programming with Java*. USA: Mc Graw Hill
12. Wu, T. (2008) *Introducción a la Programación Orientada a Objetos con Java*. España: Mc GrawHill