

 Universidad de Medellín <small>Ciencia y Libertad</small>	DISEÑO MICROCURRICULAR		Código: F-GAC-03D
			Versión: 01
			Edición: 22/08/2007
Nombre del Programa	PROGRAMACIÓN EN JAVA NIVEL I		
Facultad articulada:	Ingenierías		
Tipo de Evento:	Curso		
Duración:	40 horas		
Justificación: Desde el punto de vista laboral, el sector informático está demandando cada vez más ingenieros, tecnólogos o cualquier tipo de analista que tenga conocimientos en la programación orientada a objetos, aplicada sobre tecnologías JAVA y J2EE, por esta razón, es de gran importancia divulgar este tipo de conocimiento con el fin de preparar y mejorar la demanda de mano de obra calificada.			
Objetivo general: Introducir a los participantes en el mundo de uno de los lenguajes de programación más importantes en el desarrollo de software orientado a objetos, que ha generado no solo controversia por su portabilidad si no que ha marcado un punto de referencia en el desarrollo de aplicaciones para Internet. Además de conocer los fundamentos necesarios para desarrollar aplicaciones mediante la comprensión de sus principales APIs.			
Objetivos específicos: <ul style="list-style-type: none">• Desarrollar las habilidades de programación en JAVA• Entender el esquema de programación orientados a objetos de JAVA• Ofrecer herramientas comunes de programación y utilidades de JAVA existentes.			
Competencias: <ul style="list-style-type: none">• Desarrollo de aplicaciones orientadas a objetos• Análisis de problemas orientados a la programación JAVA• Manejo de un entorno de desarrollo IDE			
Público objetivo: Tecnólogos, técnicos, estudiantes de Ingeniería, estudiantes de Ingeniería de Sistemas y en general personas que les guste la programación y estén interesadas en aprender del mundo de Java y la programación orientada a objetos.			
Conocimientos previos requeridos: <ul style="list-style-type: none">• Conocimientos básicos en sistemas operativos• Lógica de programación básica			

ESTRUCTURA		
SESIÓN	TEMAS Y SUBTEMAS	INTENSIDAD HORARIA
	Java como lenguaje: 1.1. ¿Qué es Java? 1.2. Filosofía de Java 1.3. Historia del lenguaje 1.4. Características 1.5. ¿Por qué Java? 1.6. ¿Cómo funciona Java?	2 horas
	Uso básico de IntelliJ community e IDE'S Afines: 2.1. Qué es un IDE 2.2. Espacios de trabajo y proyectos 2.3. Paquetes de clases 2.4. Creación de Aplicaciones 2.5. Creación de Proyectos 2.6. Configuración del entorno 2.7. Opciones de personalización del IDE	2 horas
	Instrucciones, declaraciones y control de acceso 3.1. Identificadores, Operadores, Tipos de datos 3.1. Declaración de variables y métodos 3.2. Palabras claves o reservadas 3.3. Modificadores de acceso 3.4. Paquetes e importaciones 3.5. Enumeraciones 3.6. Constantes	4 horas
	Java Orientado a Objetos: 4.1. Conceptos básicos de objetos 4.2. Abstracción 4.3. Encapsulación 4.4. Polimorfismo 4.5. Sobrecarga 4.6. Herencia 4.7. Composición: Relaciones HAS-A, IS-A 4.8. Redefinición de Métodos 4.9. Casting 4.10. Declaración de Interfaces 4.11. Uso de constructores 4.12 Acoplamiento y cohesión	6 horas
Caso de estudio	Ejemplo: <i>Electrodomésticos</i>	4 horas

	<p>Patrones y principios de desarrollo:</p> <ul style="list-style-type: none">5.1. Patrones de diseño5.2. Patrones GRASP5.3. Principios SOLID.5.4. Ejemplos y ejercicios en java.	<p>4 horas</p>
--	---	-----------------------

	Estructuras condicionales y bucles 6.1 Bucles en java (for while) 6.2 Sentencia Switch 6.3 Estructuras Condicionales Java: if, if-else, switch, break, continue, 6.4 Ejemplos con estructuras de datos simples.	4 horas
	Colecciones en java. 7.1 ¿Por qué usar colecciones? 7.2 Jerarquía de colecciones 7.3 Listas y array. 7.4 Mapas y sets. 7.5 Ventajas y desventajas entre colecciones. 7.6 Ejemplos prácticos.	4 horas
	Manejo de excepciones: 8.1 ¿Qué es una excepción? 8.2 Herencia de excepciones y las clases involucradas 8.3 Tipos de excepciones. 8.4 Excepciones personalizadas 8.5 Excepciones chequeadas y no chequeadas 8.6 Explicación del bloque Try, catch, finally. 8.7 Múltiples catch con excepciones	4 horas
	Asignaciones y variables: 9.1 Referencias y uso de Memoria: Stack y Heap 9.2 Determinar los efectos de pasar variables entre métodos 9.3 Distinguir cuando los objetos son reconocidos por el recolector de basura de la máquina virtual (Garbage Collector) para liberar recursos	2 horas
	Paquete IO: 10.1 Streams 10.2 Byte Streams 10.3 Character Streams 10.4 Clases Reader - Writer 10.5 Clase RandomAccessFile 10.6 Clase File	2 horas
	Paquete Lang: 11.1 La clase Math 11.2 La clase Character 11.3 Utilización String y StringBuffer	2 horas
METODOLOGÍA		

El curso Java para Construcción de Aplicaciones está diseñado con las siguientes características:

- Se entregará bibliografía y resumen de cada sesión, en formato digital.
- Se hará énfasis en la primera parte a los conceptos teóricos para introducir a la práctica mediante ejercicios.
- En el contenido del curso existen varios temas transversales, los cuales se verán en transcurso de las clases.
- En cada una de las clases se realizarán ejemplos prácticos para aplicar los conocimientos adquiridos.