

Nombre del Programa

**PROGRAMACIÓN EN JAVA AVANZADO**

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:**

- 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.
- Profundizar en las diferentes utilidades de Java.
- Aprender de patrones de diseño y cómo aplicarlos.
- Desarrollar un módulo completo utilizando los componentes explicados en el curso.

**Competencias:**

- Desarrollo de aplicaciones orientadas a objetos.
- Análisis de problemas orientados a la programación JAVA.
- Patrones de diseño.

**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:**

- Conocimientos básicos en sistemas operativos
- Conocimientos de Java nivel básico
- Conocimientos de programación orientada a objetos

**ESTRUCTURA**

<b>SESIÓN</b>	<b>TEMAS Y SUBTEMAS</b>	<b>INTENSIDAD HORARIA</b>
	<b>Introducción:</b> <ul style="list-style-type: none"><li>• Tipos de aplicación.</li><li>• <b>Deployment.</b></li><li>• Maven.</li><li>• Gradle.</li></ul>	2 horas
	<b>Collections y Generics Framework:</b> <ul style="list-style-type: none"><li>• Describir las implementaciones de propósito general de las interfaces principales en el Framework de Colecciones.</li><li>• Maps y Collections</li><li>• interfaces Comparable y Comparator</li><li>• Utilice colecciones genéricas y parámetros de tipo en clases genéricas.</li><li>• Refactorizar código no genérico existente</li><li>• Escribir un programa para iterar sobre una colección.</li></ul>	2 horas
	<b>I/O Fundamentals:</b> <ul style="list-style-type: none"><li>• Escribir un programa que utilice argumentos de línea de comandos y propiedades del sistema.</li><li>• Examinar la clase Propiedades</li><li>• Construya nodos y flujos de procesamiento, y utilícelos apropiadamente</li><li>• Serializar y deserializar objetos.</li><li>• Distinguir lectores y escritores de las corrientes y seleccionar adecuadamente entre ellos.</li><li>• Leer datos de la consola</li><li>• Escribir datos en la consola.</li><li>• Describir archivos y E/S de archivos.</li></ul>	4 horas

	<b>Hilos y Concurrencia:</b> <ul style="list-style-type: none"> <li>• Definir un hilo</li> <li>• Crear subprocesos separados en un programa Java, controlando el código y los datos que utiliza ese subproceso.</li> <li>• Controlar la ejecución de un hilo y escribir código independiente de la plataforma con hilos</li> <li>• Utilice esperar y notificar para comunicarse entre hilos</li> <li>• Utilice sincronizado para proteger los datos de la corrupción.</li> <li>• Utilice diferentes comportamientos predefinidos para los hilos y concurrencia.</li> </ul>	4 horas
	<b>Networking:</b> <ul style="list-style-type: none"> <li>• Desarrollar código para configurar conexiones de red.</li> <li>• Entender TCP/IP</li> <li>• Utilice las clases ServerSocket y Socket para implementar clientes y servidores TCP/IP</li> </ul>	2 horas
	<b>Inner Class y Expresiones lambda:</b> <ul style="list-style-type: none"> <li>• Entender los diferentes tipos de inner clases.</li> <li>• Interfaces funcionales.</li> <li>• ParallelStream</li> <li>• Utilice todos los tipos de interfaces funcionales.</li> <li>• Personalice sus propias interfaces funcionales.</li> </ul>	2 horas
	<b>Patrones de diseño:</b> <ul style="list-style-type: none"> <li>• Creacionales</li> <li>• Estructurales</li> <li>• Comportamiento</li> </ul>	4 horas
	<b>Modularity</b> <ul style="list-style-type: none"> <li>• Tipos</li> <li>• Componentes</li> <li>• Paquetes y recursos</li> </ul>	2 horas
	<b>JDBC</b> <ul style="list-style-type: none"> <li>• Operaciones básicas</li> <li>• Invocación de procesos</li> <li>• Transacciones</li> </ul>	4 horas

	<ul style="list-style-type: none"> <li>• Pool de Conexiones</li> </ul>	
	<b>JPA</b> <ul style="list-style-type: none"> <li>• Genérico</li> <li>• Hibernate</li> <li>• JPQL / HQL</li> </ul>	2 horas
	<b>Servlets</b> <ul style="list-style-type: none"> <li>• Conceptos básicos</li> <li>• Tipos de operaciones</li> <li>• Ciclo de vida</li> <li>• Request y Response</li> </ul>	4 horas
	<b>Web Services</b> <ul style="list-style-type: none"> <li>• Tipos de Web Services en Java EE</li> <li>• Operaciones.</li> <li>• Comunicación entre procesos.</li> <li>• Diferencias entre Rest y SOAP.</li> <li>• Seguridad (tentativo)</li> </ul>	4 horas
	<b>Proyecto práctico</b>	4 horas

## **METODOLOGÍA**

- 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.
- Cada estudiante construirá su perfil de trabajo colaborativo en cuanto a repositorio de código.