



<b>Servicio de Instrucción</b>	Oficialización: Director Edwin Maraví
CURSO <b>PROGRAMMER 8</b> JAVA	Año:2018
	Código : <b>C01_2018</b>

## Sílabo Programmer 8

### I. DATOS GENERALES

Año de vigencia	:	2018
Número de sesiones	:	20 (4 horas por sesión)
N° de Horas académicas	:	80
Requisitos	:	- Conocimientos básicos de programación. - Conocimientos básicos de bases de datos. - Conocimientos básicos de diseño Web.

### II. FUNDAMENTACIÓN

La tecnología Java viene evolucionando rápidamente y de alguna manera ha vuelto al concepto original que era ser usado por cualquier dispositivo. El nuevo paradigma de Internet de las Cosas, es el escenario perfecto para poner en práctica todo lo que Java puede hacer. Toda aplicación necesita almacenar su información en un repositorio estático de datos que pueden ser archivos o base de datos dando la capacidad de poder usar de forma transparente una capa de presentación de escritorio o web según las necesidades de la solución. Este curso orienta a los participantes al correcto uso de las librerías de Java y las buenas prácticas en el momento del desarrollo de aplicaciones en tres capas, usando patrones de diseño de software.

### III. UNIDAD DE COMPETENCIA

Desarrollar componentes de software usando la tecnología Java SE y Java EE.

Tener preparación necesaria para examen de certificación OCPJP 8 (Java SE 8 Programmer II 1Z0-809) IV.

### CAPACIDADES TERMINALES Y CRITERIOS DE EVALUACIÓN

CAPACIDAD TERMINAL	CRITERIOS DE EVALUACIÓN
Desarrollar aplicaciones de escritorio y web usando componentes y librerías especializadas en Java.	Fundamentos del lenguaje
	Programación orientada a objetos
	Patrones de diseño de software
	Expresiones y operaciones Lambdas
	Excepciones y Aserciones.
	Hilos y Concurrencia
	Persistencia de Datos: Archivos y Bases de datos.
	Desarrollo de proyectos web usando MVC.



<b>Servicio de Instrucción</b>	Oficialización: Director Edwin Maraví
CURSO <b>PROGRAMMER 8</b> JAVA	Año:2018
	Código : <b>C01_2018</b>

## V. CONTENIDOS

Sesión	Contenido
<b>Sesión 01</b>	<p><u>Revisión de la plataforma Java.</u> Definición de cómo el lenguaje java se hace independiente de la plataforma. Diferencia entre las plataformas Java ME, Java SE y Java EE. Importancia del Java Community Process.</p> <p><u>Sintaxis Java y revisión de clases.</u> Creación de clases simples. Creación de variables primitivas. Uso de Operadores. Convenciones de nombres en Java. Creación y manipulación de cadenas. Usando sentencias if-else y switch. Iteraciones con loops: while, do-while, for y for mejorado. Creación de arreglos. Usando campos, constructores y métodos Java. Pasando datos por valor y por referencia. Uso de paquetes. Java Class Loader y Garbage Collector.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S01.PDF (archivo disponible en aula virtual). <b>Desarrollo de una solución orientada objetos académica.</b></li> </ul>
<b>Sesión 02</b>	<p><u>Encapsulamiento y Subclases.</u> Uso de encapsulamiento en diseño de clases Java. Modelo de lógica de negocio usando clases Java. Clases inmutables. Creación de clases y sub clases. Sobre carga de métodos.</p> <p><u>Sobre escritura de métodos, Polimorfismo y Clases estáticas.</u> Uso de niveles de acceso: privado, protegido, por defecto y público. Sobre escrituras de métodos. Uso de invocación virtual de métodos. Uso de varargs para especificar variables argumentos. Casting y sus reglas. Modelando problemas de negocio usando la palabra reservada static. Implementando el patrón de diseño Singleton.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S02.PDF (archivo disponible en aula virtual). <b>Desarrollo de una calculadora orientada a objetos.</b></li> </ul>
<b>Sesión 03</b>	<p><u>Clases abstractas y anidadas.</u> Diseño de clases base de propósito general usando clases abstractas. Construcción de clases y subclases abstractas java. Aplicación de la palabra reservada final. Distinción entre clases de alto nivel y anidadas.</p> <p>Interfaces y expresiones Lambda. Definición de una interface. Elección entre herencia entre clases e interfaces. Extensión de una interface. Clases internas anónimas. Definición de una Expresión Lambda.</p> <p><u>Colecciones y Genéricos.</u> Creación de una clase genérica personalizada. Uso del tipo inferido diamante para crear un objeto. Creación de colecciones usando genéricos. Implementación de ArrayList, TreeSet, HashMap y Deque. Ordenando colecciones. AutoBoxing y UnBoxing. <u>Desarrollo con patrones Java.</u> Implementación de patrón DAO. Necesidad de Patrones en Desarrollo de Software. DAO y Factory juntos.</p> <p><u>Poniendo todo junto.</u> Diagrama de clases de caso propuesto.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S03.PDF (archivo disponible en aula virtual). <b>Desarrollo de caso comercial.</b></li> <li>▪ <b>Publicar el primer trabajo de investigación.</b></li> </ul>



<b>Servicio de Instrucción</b>	Oficialización: Director Edwin Maraví
<b>CURSO</b>  <b>JAVA</b>  <b>PROGRAMMER 8</b>	Año:2018
	Código : <b>C01_2018</b>

<b>Sesión</b> <b>04</b>	<p><u>Procesamiento de cadenas</u>. Sub cadenas. Expresiones regulares. Properties. Parseando cadenas. La clase StringTokenizer. Pattern y Matcher. La clase StringBuilder.</p> <p><u>Collections Streams, y Filters</u>. Descripción del patrón Builder. Iterando a través de una colección usando sintaxis Lambda. Descripción de la interface Stream. Filtro de una colección usando expresiones lambda. Llamado de un método existente usando una referencia de métodos. Encadenando múltiples métodos. Definición de túneles (pipelines) en términos de lambdas y colecciones.</p>
----------------------------	---

	<p><u>Interfaces Funcionales Lambda internas</u>. Lista de interfaces internas incluidas en java.util.function. Interfaces Core: Predicate, Consumer, Function, Supplier. Uso de versiones primitivas de interfaces base. Uso de versiones binarias de interfaces base.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S04.PDF (archivo disponible en aula virtual). <b>Desarrollo de caso financiero usando diagrama de clases y patrones de diseño de software.</b></li> <li>▪ <b>Entregar la primera tarea virtual.</b></li> </ul>
<b>Sesión</b> <b>05</b>	<p><b>Primera evaluación práctica</b></p> <ul style="list-style-type: none"> <li>▪ <b>Desarrollar del primer cuestionario virtual.</b></li> </ul>
<b>Sesión</b> <b>06</b>	<p><u>Operaciones Lambda</u>. Extracción de datos de un objeto usando map. Descripción de tipos de operaciones con streams. Descripción de la clase Optional. Descripción de procesamiento LAZY. Ordenando un stream. Grabado de resultados a una colección usando el método collect. Agrupando y partiendo datos usando la clase Collector.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S06.PDF (archivo disponible en aula virtual). <b>Desarrollo de caso comercial usando colecciones.</b></li> </ul>
<b>Sesión</b> <b>07</b>	<p><u>Exceptions y Assertions</u>. Definición de propósito de las excepciones. Capturando y extendiendo excepciones. El operador Try Catch y MultiCatch. La cláusula finally. Suprimiendo excepciones. Interface AutoClosable.</p> <p><u>Pruebas Unitarias</u>. Uso de JUnit. Probando invariants usando assertions.</p> <p><u>API Java Date/Time</u>. Creación y manejo de eventos basados en fechas. Creación y manejo de eventos basados en tiempo. Combinando fecha y tiempo en una misma clase. Trabajo con fechas y tiempos a través de zonas. Administración de cambios resultantes de cambios de días. Definición y creación de timestamps, periodos y duraciones. Aplicación de formatos a fechas y tiempos locales y de zona.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S07.PDF (archivo disponible en aula virtual). <b>Desarrollo de caso académico usando patrones y excepciones a su vez se probando el código generado con pruebas unitarias.</b></li> </ul>
<b>Sesión</b> <b>08</b>	<p><u>Fundamentos de Java IO</u>. Escribiendo y leyendo aplicaciones de consola. Datos con Streaming. Proceso de serialización y des serialización. Clase File, BufferedReader, PrintWriter. Lectura y escritura en archivos planos.</p> <p><u>Fundamentos de Java File (NIO.2)</u>. Gestión de archivos. Copia Recursiva. PathMatcher, manejo de rutas lógicas en unidades.</p>



<b>Servicio de Instrucción</b>	Oficialización: Director Edwin Maraví
<b>CURSO</b>  <b>JAVA</b>  <b>PROGRAMMER 8</b>	Año:2018
	Código : <b>C01_2018</b>

	<ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S08.PDF (archivo disponible en aula virtual). <b>Desarrollo de caso académico usando archivos.</b></li> <li>▪ <b>Publicar el segundo trabajo de investigación.</b></li> </ul>
<b>Sesión 09</b>	<p><u>Hilos</u>. Manejo de tareas. La clase Thread y la interface Runnable. Problemas con datos compartidos. Operaciones Atómicas. Manejando interrupciones. Sincronizando acceso a datos compartidos. Implementando un programa multi-hilos.</p> <p><u>Concurrencia</u>. Usando el paquete java.util.concurrent, atomic y locks. Callable ExecutorService. El framework Fork-Join. Manejo del paralelismo. Necesidad de Fork-Join. Tareas recursivas. <u>Streams Paralelos</u>. Revisión de las principales características de streams. Descripción de cómo hacer para ejecutar un flujo de streams en paralelo. Definición de reducción.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S09.PDF (archivo disponible en aula virtual). <b>Simulación de programación de tareas en Java.</b></li> </ul>

	<ul style="list-style-type: none"> <li>▪ <b>Entregar la segunda tarea virtual.</b></li> </ul>
<b>Sesión 10</b>	<p><b>Segunda evaluación práctica</b></p> <ul style="list-style-type: none"> <li>▪ <b>Desarrollar el segundo cuestionario virtual.</b></li> </ul>
<b>Sesión 11</b>	<p>Construcción de aplicaciones con Base de Datos con JDBC. Componentes de la API JDBC. Manejo de excepciones JDBC. ResultSet. Trabajando con Mysql y Postgress. Usando el patrón Data Access Object. Uso de RowSet.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S11.PDF (archivo disponible en aula virtual). <b>Probar con JUnit consultas de motores de bases de datos distintos.</b></li> </ul>
<b>Sesión 12</b>	<p><u>Sentencias SQL con Java</u>. Diferencia entre StateMent y PreparedStatement. Manejo de Procedimientos almacenados usando CallableStatement.</p> <p><u>Manejo de Cursores con Java</u>.</p> <p><u>Administración de Sucesos con Java</u>. Logs para java con Log4J.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S12.PDF (archivo disponible en aula virtual). <b>Caso de mantenimiento de tablas de bases de datos.</b></li> </ul>
<b>Sesión 13</b>	<p>Manejando Transacciones (ACID). Transacciones JDBC. <u>Localización</u>. Descripción de ventajas de la localización.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S13.PDF (archivo disponible en aula virtual). <b>Caso de manejo de transacciones en un banco.</b></li> <li>▪ <b>Publicar el tercer trabajo de investigación.</b></li> </ul>
<b>Sesión 14</b>	<p><u>Java Foundation Classes</u>. Swing API. Controladores y componentes Swing. Manejo de eventos en componentes Swing.</p> <p><u>Poniendo todo junto</u>. Aplicación Swing con conexión de base de datos usando patrones de Diseño y manejo de Transacciones.</p> <ul style="list-style-type: none"> <li>▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S14.PDF (archivo disponible en aula virtual). <b>Desarrollo de caso de cuentas bancarias con Swing.</b></li> <li>▪ <b>Desarrollar la tercera tarea virtual.</b></li> </ul>



<b>Servicio de Instrucción</b>	Oficialización: Director Edwin Maraví
<b>CURSO</b>  <b>JAVA</b>  <b>PROGRAMMER 8</b>	Año:2018
	Código : <b>C01_2018</b>

<b>Sesión 15</b>	<b>Tercera evaluación práctica</b> ▪ <b>Desarrollar el tercer cuestionario virtual.</b>
<b>Sesión 16</b>	<u>Introducción a las tecnologías de aplicaciones Web.</u> Aplicaciones Web y Servidores de Aplicaciones. Arquitectura de aplicaciones Java EE. <u>Plataforma Java EE.</u> Especificación Servlet y su ciclo de vida. Especificación JSP y su ciclo de vida. <u>Definición de Arquitectura Modelo Vista Controlador (MVC)</u> ▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S16.PDF (archivo disponible en aula virtual). <b>Proyecto login en versión Web.</b>
<b>Sesión 17</b>	<u>Ciclo de vida de una aplicación Web.</u> Despliegue de una aplicación. Declarando páginas de inicio. Configurando Parámetros de contexto. Mapeando errores y pantallas de error. Declarando referencias de recursos. ▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S17.PDF (archivo disponible en aula virtual). <b>Desarrollo de una Intranet usando Java EE, JDBC y MVC.</b>
<b>Sesión 18</b>	<u>Generación de reportes web.</u> Reportes tipo gráfico de barras, lineales y de pie con JFreeChart. Reportes con iReport y JasperReport. Generación de documentos MS Excel, PowerPoint, Word, Visio, Publisher, Outlook. Generación de reportes PDF. ▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S18.PDF (archivo disponible en aula virtual). <b>Desarrollo de reportes web para caso de Académico.</b>

	▪ <b>Publicar el cuarto trabajo de investigación.</b>
<b>Sesión 19</b>	<u>Gestión de Proyectos.</u> Aprendiendo a usar MAVEN. Configuración de dependencias con archivos POM. <u>Poniendo todo junto:</u> Patrones: Dao Factory- Singleton-MVC, JDBC: Stored Procedures, Pruebas: JUnit, Reportes: IReport. ▪ Desarrollar los ejercicios propuestos en el documento EJERCICIOS_S19.PDF (archivo disponible en aula virtual). <b>Intranet con acceso de usuarios y mantenimiento de tablas con consultas y reportes Web.</b> ▪ <b>Entregar la cuarta tarea virtual.</b>
<b>Sesión 20</b>	<b>Cuarta evaluación práctica</b> ▪ <b>Desarrollar el cuarto cuestionario virtual.</b>

## VI. METODOLOGÍA

- Curso teórico-práctico.
- Exposición del profesor, ayudas audiovisuales, diálogo abierto y debate.
- Realización de casos prácticos en laboratorio.
- Realización de tareas en aula virtual.
- Generación de conocimiento mediante investigación y foros de discusión en aula virtual.



<b>Servicio de Instrucción</b>	Oficialización: Director Edwin Maraví
<b>CURSO</b> <b>JAVA</b> <b>PROGRAMMER 8</b>	Año:2018
	Código : <b>C01_2018</b>

#### VII. EVALUACIÓN

La evaluación, como constatación del logro de los objetivos que es alcanzar las capacidades mencionadas se dará mediante la siguiente ponderación:

04 Tareas (se elimina la menor)	<b>15 %</b>
04 Evaluaciones Virtuales (se elimina la menor)	<b>10 %</b>
04 Investigaciones (se elimina la menor)	<b>15 %</b>
Evaluaciones prácticas 1 y 2 (se elimina la menor)	<b>20 %</b>
Evaluaciones prácticas 3 y 4	<b>40 %</b>
<b>TOTAL</b>	<b>100 %</b>

#### VIII. BIBLIOGRAFÍA

- DIETEL & DIETEL - La Biblia de Java.
- OFALI, ROBERT y HARKEY, DAN - Cliente Servidor. McGraw-Hill, 2da Edición, México, 2001
- GALLAGHER, SIMON y HERBERT, SIMON - Cliente Servidor e Internet Addison Wesley, 2da Edición. México, 2002

#### IX. PÁGINAS WEB DE APOYO:

- <http://www.programacion.com/java>
- <http://www.javahispano.com>
- <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>
- <http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>
- <http://www.elvex.ugr.es/decsai/java/pdf/>
- <http://www.cjavaperu.org>