

# UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

COORDINACIÓN DE FORMACIÓN BÁSICA

COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA

## PROGRAMA DE UNIDAD DE APRENDIZAJE HOMOLOGADO

### I. DATOS DE IDENTIFICACIÓN

1. Unidad académica (s): **CAMPUS ENSENADA:** FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO, FACULTAD DE INGENIERÍA Y NEGOCIOS SAN QUINTÍN  
**CAMPUS MEXICALI:** FACULTAD DE INGENIERÍA, ESCUELA DE INGENIERÍA Y NEGOCIOS GUADALUPE VICTORIA  
**CAMPUS TIJUANA:** FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA, FACULTAD DE INGENIERÍA Y NEGOCIOS
2. Programa (s) de estudio: (Técnico, Licenciatura (s)): INGENIERÍA EN COMPUTACIÓN
3. Vigencia del plan: 2009-2
4. Nombre de la unidad de aprendizaje Programación Orientada a Objetos Avanzada
5. Clave: 12105
6. HC:      HL: 2 HT: 2 HPC:      HCL:      HE      CR 4
7. Etapa de formación a la que pertenece: Disciplinaria
8. Carácter de la unidad de aprendizaje: Obligatoria X Optativa
9. Requisitos para cursar la unidad de aprendizaje : Programación Orientada a Objetos

## **II. PROPÓSITO GENERAL DEL CURSO**

Que el alumno utilice herramientas del API de Java que le permitan programar aplicaciones gráficas orientadas a objetos, que sean eficientes y hagan uso tanto de recursos contenidos en la computadora donde se ejecuta la aplicación, como de recursos localizados en el contexto de una red de computadoras.

## **III. COMPETENCIA (S) DEL CURSO**

Desarrollar sistemas orientados a objetos aplicando técnicas de reutilización de código para solucionar de forma ordenada y eficiente los problemas de procesamiento de información.

## **IV. EVIDENCIA (S) DE DESEMPEÑO**

Realizar una aplicación de cómputo basado en la reutilización de componentes mediante una interfaz gráfica utilizando el paradigma orientado a objetos

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

Aplicar de forma eficiente y ordenada los mecanismos que tiene java para agrupar clases, buscando mantener la reutilización de código.

### CONTENIDO HORAS

**DURACIÓN 4**

- Unidad 1 Paquetes de Java
- 1.1 Definición de un paquete en Java
- 1.2 Ventajas y desventajas
- 1.3 Estructura de directorios asociados
- 1.4 Compilar y ejecutar clases dentro de paquetes
- 1.5 Paquetes anidados
-

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

Utilizar los elementos necesarios para construir de forma eficiente, limpia y ordenada un entorno visual sencillo, que permita la comunicación entre los usuarios y la aplicación en java.

### CONTENIDO HORAS

**DURACIÓN 9**

#### Unidad 2 Interfaz gráfica de usuario

##### 2.1 Elementos básicos de awt

2.1.1 Componentes

2.1.2 Contenedores

##### 2.2 Paquete javax.swing

2.2.1 Ventanas

2.2.2 JFrame

2.2.2.1 JPanel

2.2.2.2 JOptionPane

2.2.2.3 JFileChooser

2.2.2.4 JMenu

2.2.3 Componentes de captura y despliegue

2.2.3.1 JLabel

2.2.3.2 JTextArea

2.2.3.3 JTextField

2.2.4 Componente de control

2.2.4.1 JButton

2.2.4.2 Eventos del botón

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

Entender el propósito y utilizar de la manera correcta cada uno de los distintos flujos de E/S incluidos en el API de Java para la lectura y escritura de datos.

### CONTENIDO HORAS

**DURACIÓN 6**

#### Unidad 3 Flujo de Entrada y Salida

3.1 Introducción.

3.2 Flujos de bytes

3.3 Flujos de caracteres.

3.4 Entrada y salida con buffer

3.5 Lectura y escritura de archivos

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

Programar aplicaciones de cómputo que utilicen múltiples hilos para la ejecución de tareas de manera simultánea respetando la integridad de los datos utilizados.

### CONTENIDO HORAS

**DURACIÓN 4**

Unidad 4 Hilos.  
4.1 Descripción.  
4.2 Creación de hilos.  
4.3 Inicio de un hilo.  
4.4 Ejecución del hilo.  
4.5 Finalizando hilos.  
4.6 Sincronización.

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

Programar aplicaciones de cómputo que utilicen recursos disponibles en entornos de red, eligiendo el protocolo adecuado al tipo de aplicación.

### CONTENIDO HORAS

**DURACIÓN 9**

Unidad 5 Desarrollo de aplicaciones para redes

5.1 URL

5.1.1 Parámetros de los URL

5.1.2 Obteniendo el contenido de un URL

5.1.3 Enviando información a un URL

5.2 Sockets (TCP)

7.2.1 Cliente.

7.2.2 Servidor.

5.3 Datagramas (UDP)

## VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Conocer la forma de operar el laboratorio, utilizando las cuentas de usuario con actitud positiva y con responsabilidad. Preparar el entorno de trabajo personal instalando el software que utilizarán durante el semestre.	El alumno utilizará su nueva cuenta de usuario para tener acceso al equipo de cómputo y comprobar que se tiene el software necesario para trabajar durante todas las sesiones de clase.	Equipo de cómputo con JDK instalado y práctica #1 impresa.	2 horas
2	Aplicar de forma eficiente y ordenada los mecanismos que tiene java para agrupar clases, buscando mantener la reutilización de código. .	Desarrollar e implementar diferentes aplicaciones en java que serán incluidas en paquetes para aprender la forma de organizar, compilar y ejecutar dichas aplicaciones.	Equipo de cómputo con JDK y práctica #2 impresa.	4 horas
3	Utilizar los elementos necesarios para construir de forma eficiente, limpia y ordenada un entorno visual sencillo, que permita la comunicación entre los usuarios y la aplicación en java.	El alumno elaborará diferentes aplicaciones en las que se utilice la programación orientada a eventos para un ambiente gráfico para que aplique los principales componentes que proporciona el API de java.	Equipo de cómputo con JDK y práctica #2 impresa.	8 horas
4	Utilizar las clases del API de Java para entrada y salida de datos. Identificar las clases orientadas a E/S de caracteres y	El alumno programará una aplicación que realice acciones de entrada y salida básicas sobre archivos, valiéndose de una interfaz gráfica para el despliegue	Computadora con el JDK instalado y acceso a la documentación del	4 horas



	las de datos binarios y entender la forma de utilizarlas en conjunto.	de la información.	API de Java.	
5	Programar aplicaciones que realicen tareas concurrentes por medio del uso de hilos. Identificar las partes de la aplicación que pueden ejecutarse de manera simultánea.	Diseñar y programar una aplicación gráfica que ejecute varias tareas simultaneas.	Computadora con el JDK instalado y acceso a la documentación del API de Java.	4 horas
6	Identificar la estructura de un URL y obtener información de un recurso en Internet a través de una conexión.	<ul style="list-style-type: none"> <li>Se programará una aplicación que permita construir un objeto que represente un URL y que obtenga información relacionada con el recurso a través de una conexión.</li> </ul>	Computadora con el JDK instalado y acceso a la documentación del API de Java.	2 horas
7	Utilizar el modelo cliente-servidor para programar aplicaciones en red. Definir un protocolo de comunicaciones y utilizarlo para enlazar dos computadoras a través de una aplicación en red.	Definirá un protocolo para una aplicación específica y programará las dos partes de una aplicación cliente-servidor, permitiendo a varios clientes conectarse de manera simultanea.	Computadora con el JDK instalado y acceso a la documentación del API de Java.	8 horas

## VII. METODOLOGÍA DE TRABAJO

El alumno realizará lecturas previas.

Los temas serán presentados por el docente en clase, ejemplificando los conceptos mediante código elaborado en lenguaje Java.

El alumno resolverá ejercicios prácticos aplicando los conceptos vistos en clase y se utilizarán técnicas grupales buscando siempre la participación activa de los alumnos.

## VIII. CRITERIOS DE EVALUACIÓN

CALIFICACIÓN	ACREDITACIÓN	EVALUACIÓN																		
<p>La calificación final se compondrá de la siguiente manera:</p> <table><tr><td>Primer parcial ( teórico-práctico)</td><td>20%</td></tr><tr><td>Segundo parcial ( teórico-práctico)</td><td>20%</td></tr><tr><td>Tercer parcial ( teórico-práctico)</td><td>20%</td></tr><tr><td>Tareas</td><td>10%</td></tr><tr><td colspan="2">-----</td></tr><tr><td>Prácticas de laboratorio</td><td>70%</td></tr><tr><td></td><td>30%</td></tr><tr><td colspan="2">-----</td></tr><tr><td></td><td>100%</td></tr></table>	Primer parcial ( teórico-práctico)	20%	Segundo parcial ( teórico-práctico)	20%	Tercer parcial ( teórico-práctico)	20%	Tareas	10%	-----		Prácticas de laboratorio	70%		30%	-----			100%	<p>Tener un porcentaje mínimo de 80% de asistencia.</p> <p>Tener un promedio mayor o igual a 60.</p> <p>Tener acreditado el laboratorio.</p>	<p>Al término de cada unidad se realizarán dinámicas para verificar la correcta aplicación del conocimiento.</p> <p>En el caso de los exámenes prácticos se fomentará el trabajo en quipo y se realizará una retroalimentación después de la revisión de los mismos.</p>
Primer parcial ( teórico-práctico)	20%																			
Segundo parcial ( teórico-práctico)	20%																			
Tercer parcial ( teórico-práctico)	20%																			
Tareas	10%																			
-----																				
Prácticas de laboratorio	70%																			
	30%																			
-----																				
	100%																			

## IX. BIBLIOGRAFÍA

Básica	Complementaria
<p>Core Java, Vol. 1 y 2, Eighth edition. Cay S. Horstmann, Gary Cornell. Prentice-Hall. 2007.</p> <p>Java Swing, second edition. Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole. O'Reilly. 2002.</p> <p>Java Network Programming, Third Edition. Elliotte Harold, Elliotte Rusty Harold. O'Reilly. 2004.</p> <p>Java I/O. Elliotte Harold. O'Reilly. 2006.</p>	<p>Java How to program Fourth Edition Deitel &amp; Deitel Prentice Hall, 2002</p> <p>Java con Programación Orientada a Objetos y aplicaciones en la WWW Paul S. Wang Editorial Thomson, 2000</p>