



**UNIVERSIDAD NACIONAL**

**JOSÉ FAUSTINO SÁNCHEZ CARRIÓN**

**Facultad de Ciencias**

**Escuela Profesional de Estadística e Informática**

## **SÍLABO POR COMPETENCIAS**

**CURSO: LENGUAJE DE PROGRAMACIÓN II**

**DOCENTE: Lic. RAMÍREZ MUNDACA, Flor Eonice**

# SÍLABO DE LENGUAJE DE PROGRAMACIÓN II

## I. DATOS GENERALES

<b>Línea de carrera</b>	Analista de sistemas informáticos
<b>Curso</b>	Lenguaje de Programación II
<b>Código</b>	201
<b>Horas</b>	06 HT: 02; HP: 04
<b>Ciclo</b>	III

## II. SUMILLA Y DESCRIPCIÓN DEL CURSO

### Sumilla

El curso comprende el estudio y desarrollo de los siguientes temas: Programación orientada a objetos (POO), clases derivadas, coherencia, polimorfismos, sobrecarga de operadores, patrones.

### Descripción del curso

La asignatura de Lenguaje de Programación II es de naturaleza teórica - práctica, se orienta al estudio de la programación orientada a objetos -cuyos objetivos fundamentales son: hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando su calidad y reduciendo el número de fallas y, la creación de objetos que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software- para proporcionar al alumno el conocimiento necesario para la solución de problemas, la toma de decisiones, la automatización, el desarrollo de sus capacidades intelectuales y creativas mediante el manejo eficiente de un lenguaje de programación de distribución libre, robusto e independiente de la plataforma como es el lenguaje de programación Java. Uno de los aspectos prioritarios de la universidad es formar profesionales integrales, por ello se enfatiza en la apropiación de todo el conocimiento necesario y la adquisición de valores para la formación científica y humanista de los estudiantes puesta al servicio de la comunidad.

### III. CAPACIDADES AL FINALIZAR EL CURSO

Unidad	Capacidad de la unidad didáctica	Nombre de la unidad didáctica	Semanas
I	Al iniciar el aprendizaje de los fundamentos básicos de la programación orientada a objetos, crea programas sencillos, en base a un análisis bibliográfico.	<b>Conceptos básicos de la POO. Sobrecarga.</b>	4
II	Ante la necesidad de resolver situaciones más complejas, aplica los principios de herencia y polimorfismo en la creación de programas, con precisión.	<b>Clases derivadas y polimorfismo. Clases abstractas e interfaces.</b>	4
III	Ante el requerimiento de una mejor presentación y gestión de las aplicaciones, utiliza interfaces gráficas de usuario y los controles, basándose en bibliografía especializada.	<b>Interfaces gráficas de usuario. Acceso a base de datos.</b>	4
IV	A fin de resolver problemas de diseño de las aplicaciones informáticas, describe los principales patrones usados en la POO, consultando diversas fuentes de información confiables.	<b>Patrones de diseño. El patrón Modelo/Vista/Controlador (M/V/C).</b>	4

### IV. INDICADORES DE CAPACIDADES AL FINALIZAR EL CURSO

N°	INDICADORES DE CAPACIDAD AL FINALIZAR EL CURSO
01	Explica los conceptos básicos de la programación orientada a objetos.
02	Aplica los fundamentos de la POO en el diseño de programas sencillos.
03	Utiliza adecuadamente los métodos y constructores en el diseño de proyectos de programación.
04	Ejemplifica la sobrecarga de operadores y métodos mediante aplicaciones de programación.
05	Discrimina el concepto de composición del concepto de herencia de clases.
06	Indica cómo funciona el polimorfismo de clases.
07	Comprende el uso de las clases y métodos abstractos.
08	Sustenta las ventajas de usar interfaces simples y anidadas.
09	Usa las diferentes clases del paquete estándar como String, Date, Calendar, etc.
10	Manipula clases recubridoras para crear referencias de tipos de datos primitivos usados en colecciones.
11	Crea interfaces gráficas de usuario complejas haciendo uso de las clases AWT y Swing.
12	Emplea con eficacia los elementos necesarios para crear aplicaciones en java que permita la manipulación y gestión de una base de datos en Excel.
13	Comprende la importancia de utilizar los principios de la POO.
14	Fundamenta la importancia del uso del patrón Modelo/Vista/Controlador en diversas aplicaciones informáticas y la adaptación del modelo para aplicaciones web.
15	Emplea los patrones de diseño <i>Singleton</i> , <i>Factory Method</i> y <i>Abstract Factory</i> ; reconociendo qué situación es resuelta con su aplicación.
16	Emplea los patrones de diseño <i>Strategy</i> , <i>Observer</i> y <i>Decorator</i> ; reconociendo qué situación es resuelta con su aplicación.

### V. DESARROLLO DE LAS UNIDADES DIDÁCTICAS

**Capacidad de la unidad didáctica I:** Al iniciar el aprendizaje de los fundamentos básicos de la programación orientada a objetos, crea programas sencillos, en base a un análisis bibliográfico.

UNIDAD I: Conceptos básicos de la POO. Sobrecarga.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
1	1. Conceptos básicos de la POO: clases, objetos, encapsulamiento. 2. Nociones. Coherencia.	<ul style="list-style-type: none"> <li>• Discute los fundamentos de la POO.</li> <li>• Crea proyectos informáticos sencillos.</li> </ul>	<ul style="list-style-type: none"> <li>• Participa activamente en clase.</li> <li>• Desarrolla un espíritu crítico y constructivo.</li> </ul>	<ul style="list-style-type: none"> <li>• Exposición temática con la participación activa de los alumnos.</li> <li>• Uso de la tecnología informática.</li> <li>• Entrega de trabajos y retroalimentación.</li> <li>• Intervenciones orales y debate.</li> <li>• Trabajo individual.</li> <li>• Trabajo en equipo.</li> </ul>	<ul style="list-style-type: none"> <li>• Explica los conceptos básicos de la POO.</li> <li>• Aplica los fundamentos de la POO en el diseño de programas sencillos.</li> </ul>	
2	1. Definición de métodos, tipos. 2. Constructores.	<ul style="list-style-type: none"> <li>• Introduce el uso de métodos en la codificación de programas.</li> <li>• Emplea los constructores.</li> </ul>	Muestra interés para el aprendizaje y auto gestiona su aprendizaje.		<ul style="list-style-type: none"> <li>• Utiliza adecuadamente los métodos y constructores en el diseño de proyectos de programación.</li> </ul>	
3	1. Sobrecarga de operadores. 2. Sobrecarga de métodos y constructores.	<ul style="list-style-type: none"> <li>• Formula algoritmos usando la sobrecarga de operadores.</li> <li>• Crea programas usando la sobrecarga de métodos.</li> </ul>	Reflexiona sobre la importancia de los temas realizando preguntas y buscando información.		<ul style="list-style-type: none"> <li>• Ejemplifica la sobrecarga de operadores y métodos mediante aplicaciones de programación.</li> </ul>	
4	<b>Evaluación de la unidad:</b>		<b>Conocimientos:</b> Examen práctico.		<b>Producto:</b> Trabajo práctico.	<b>Desempeño:</b> Práctica de cómputo.

**Capacidad de la unidad didáctica II:** Ante la necesidad de resolver situaciones más complejas, aplica los principios de herencia y polimorfismo en la creación de programas, con precisión.

UNIDAD II: Clases derivadas y polimorfismo. Clases abstractas e interfaces.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
5	1. Composición. 2. Extensión de clases: concepto de Herencia. 3. Polimorfismo.	<ul style="list-style-type: none"> <li>• Emplea la composición y herencia de clases para crear proyectos de software.</li> <li>• Construye programas usando polimorfismo.</li> </ul>	<ul style="list-style-type: none"> <li>• Valora la utilidad de la POO.</li> <li>• Colabora en el trabajo y desarrollo de ejercicios en aula.</li> </ul>	<ul style="list-style-type: none"> <li>• Exposición temática con la participación activa de los alumnos.</li> <li>• Uso de la tecnología informática.</li> <li>• Entrega de trabajos y retroalimentación.</li> <li>• Intervenciones orales y debate.</li> <li>• Trabajo individual.</li> <li>• Trabajo en equipo.</li> </ul>	<ul style="list-style-type: none"> <li>• Discrimina la composición de la herencia de clases.</li> <li>• Indica cómo funciona el polimorfismo de clases.</li> </ul>	
6	1. Definición de clases abstractas y métodos abstractos.	<ul style="list-style-type: none"> <li>• Desarrolla programas utilizando el concepto de clase abstracta.</li> <li>• Formula métodos abstractos.</li> </ul>	Acepta ideas de los demás y expone sus propias ideas.		<ul style="list-style-type: none"> <li>• Comprende el uso de las clases y métodos abstractos.</li> </ul>	
7	1. Interfaces. 2. Clases e interfaces anidados.	<ul style="list-style-type: none"> <li>• Compara la eficacia de utilizar una clase abstracta o una interfaz.</li> <li>• Desarrolla en forma grupal las preguntas propuestas.</li> </ul>	Asume el trabajo con responsabilidad y espíritu crítico.		<ul style="list-style-type: none"> <li>• Sustenta las ventajas de usar interfaces.</li> <li>• Usa interfaces anidadas.</li> </ul>	
8	<b>Evaluación de la unidad:</b>		<b>Conocimientos:</b> Examen práctico.		<b>Producto:</b> Guía de Problemas.	<b>Desempeño:</b> Intervenciones orales.

**Capacidad de la unidad didáctica III:** Ante el requerimiento de una mejor presentación y gestión de las aplicaciones, utiliza interfaces gráficas de usuario y los controles, basándose en bibliografía especializada.

UNIDAD III: Interfaces gráficas de usuario. Acceso a base de datos.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
9	1. Algunas clases del paquete estándar. 1.1. Cadenas. Clases recubridoras. 1.2. Colecciones. 1.3. La clase Date, Calendar, etc.	<ul style="list-style-type: none"> <li>• Discute la utilidad de usar la clase Sting Date, Calendar, etc.</li> <li>• Identifica qué es una colección.</li> </ul>	Demuestra buena disposición para el aprendizaje y el trabajo en aula.	<ul style="list-style-type: none"> <li>• Exposición temática con la participación activa de los alumnos.</li> <li>• Uso de la tecnología informática.</li> <li>• Entrega de trabajos y retroalimentación.</li> <li>• Intervenciones orales y debate.</li> <li>• Trabajo individual.</li> <li>• Trabajo en equipo.</li> </ul>	Usa las diferentes clases del paquete estándar como String, Date, Calendar, etc; las clases recubridoras y las colecciones.	
10	1. Interfaces gráficas de usuario. 1.1. Contenedores y componentes. 1.2. Eventos.	<ul style="list-style-type: none"> <li>• Identifica un contenedor de un componente.</li> <li>• Codifica diversos eventos.</li> </ul>	Participa en clase exponiendo y debatiendo sus ideas.			Crea interfaces gráficas de usuario complejas usando las clases AWT y Swing.
11	1. Acceso a base de datos en Access. 2. Ingreso y consulta de datos. 3. Instrucciones SQL.	<ul style="list-style-type: none"> <li>• Construye aplicaciones con acceso a base de datos.</li> <li>• Utiliza las diferentes sentencias SQL.</li> </ul>	<ul style="list-style-type: none"> <li>• Muestra superación.</li> <li>• Acepta ideas de los demás y hace un análisis crítico.</li> </ul>			Emplea los elementos necesarios para crear aplicaciones en java que permita la gestión de una base de datos en Excel.
12	<b>Evaluación de la unidad:</b>	<b>Conocimientos:</b> Práctica dirigida.		<b>Producto:</b> Trabajo de investigación.	<b>Desempeño:</b> Debate.	

**Capacidad de la unidad didáctica IV:** A fin de resolver problemas de diseño de las aplicaciones informáticas, describe los principales patrones usados en la POO, consultando diversas fuentes de información confiables.

UNIDAD IV: Patrones de diseño. El patrón Modelo/Vista/Controlador.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
13	1. Principios de la POO. 2. Concepto de patrones de diseño. 3. El patrón M/V/C.	<ul style="list-style-type: none"> <li>• Discute los principios de la POO.</li> <li>• Discute la importancia del uso de los distintos patrones de diseño tal como M/V/C.</li> </ul>	<ul style="list-style-type: none"> <li>• Asume el trabajo con buena actitud.</li> <li>• Colabora con sus demás compañeros.</li> </ul>	<ul style="list-style-type: none"> <li>• Exposición temática con la participación activa de los alumnos.</li> <li>• Uso de la tecnología informática.</li> <li>• Entrega de trabajos y retroalimentación.</li> <li>• Intervenciones orales y debate.</li> <li>• Trabajo individual.</li> <li>• Trabajo en equipo.</li> </ul>	Comprende la importancia de utilizar los principios de la POO y el patrón M/V/C.	
14	1. Patrón <i>Singleton</i> . 2. Patrón <i>Factory Method</i> . 3. Patrón <i>Abstract Factory</i> .	Discute las ventajas de los patrones <i>Singleton</i> , <i>Factory Method</i> y <i>Abstract Factory</i> .	Fomenta un ambiente de compañerismo para discutir y analizar los problemas propuestos.			Emplea los patrones de diseño <i>Singleton</i> , <i>Factory Method</i> y <i>Abstract Factory</i> ; en la situación correcta.
15	1. Patrón <i>Strategy</i> . 2. Patrón <i>Observer</i> . 3. Patrón <i>Decorator</i> .	Discute las ventajas de los patrones <i>Strategy</i> , <i>Observer</i> y <i>Decorator</i> .	Discute sobre la importancia del tema estudiado.			Emplea los patrones de diseño <i>Strategy</i> , <i>Observer</i> y <i>Decorator</i> ; en la situación correcta.
16	<b>Evaluación de la unidad:</b>	<b>Conocimientos:</b> Evaluación escrita.		<b>Producto:</b> Proyecto de software.	<b>Desempeño:</b> Exámenes orales.	

## VI. MATERIALES EDUCATIVOS Y OTROS RECURSOS DIDÁCTICOS

Los materiales educativos y recursos didácticos que se utilizarán en el desarrollo del presente curso:

### MEDIOS ESCRITOS

- Materiales convencionales como separatas, guías de práctica, libros y referencias web.

### MEDIOS VISUALES Y ELECTRÓNICOS

- Laptop, Presentaciones multimedia, animaciones y simulaciones interactivas.

### MEDIOS INFORMÁTICOS

- Servicios telemáticos: sitios web, correo electrónico, chats, foros.

## VII. EVALUACIÓN

**Evidencias de conocimiento:** El conocimiento se evidenciará a través de exámenes prácticos, prácticas dirigidas y evaluaciones escritas.

**Evidencias de desempeño:** El desempeño se evidenciará a través de prácticas de cómputo, intervenciones orales, debates y exámenes orales.

**Evidencias de producto:** El producto se evidenciará utilizando trabajos prácticos, guías de problemas, trabajos de investigación y proyectos de software.

El sistema de evaluación se rige por el Reglamento Académico General aprobado por Resolución de Consejo Universitario N° 0105-2016-CU-UH de fecha 01 de marzo del 2016.

La evaluación es un proceso permanente e integral que permite medir el logro del aprendizaje alcanzado por los estudiantes de las Escuelas Profesionales.

El sistema de evaluación es integral, permanente, cualitativo y cuantitativo (vigesimal) y se ajusta a las características de las asignaturas dentro de las pautas generales establecidas por el Estatuto de la Universidad y el presente Reglamento (Art. 124 y 125).

*El carácter integral de la evaluación de las asignaturas comprende la Evaluación Teórica, Práctica y los Trabajos Académicos, y el alcance de las competencias establecidas en los nuevos planes de estudios.*

*Para la Evaluación de la parte teórica - práctica se podrá emplear los siguientes procedimientos e instrumentos: Prueba escrita, Individuales o grupales, prácticas calificadas de aula. Evaluación Oral con pruebas orales, exposiciones, discusiones y demostraciones.*

*Para la Evaluación mediante trabajos académicos de acuerdo a la naturaleza de cada asignatura y/o aplicativos se podrán emplear los siguientes procedimientos e instrumentos: Prácticas calificadas con guía de observación e informe, trabajos monográficos, otros trabajos Académicos (Art. 126).*

### **Control de Asistencia a Clases:**

- La *asistencia* a clases teóricas y prácticas son obligatorias. La acumulación de más del 30% de inasistencia no justificadas, dará lugar a la desaprobación de la asignatura por límite de inasistencia con nota cero (00) (art. 121)
- *La asistencia a las asignaturas es obligatoria en un mínimo de 70%, lo que dará lugar a la inhabilitación por no justificar las inasistencias, de acuerdo al art. 122, que menciona que el estudiante está obligado a justificar su inasistencia, en un plazo no mayor a tres (3) días hábiles; ante el Director de la Escuela Profesional quien derivará el documento al docente a más tardar en dos (2) días. Opcionalmente el estudiante presentará una copia del expediente de justificación al docente. (art. 122 y 123).*

### Sistema de Evaluación (Para los Currículos por Competencia):

Este curso está programado en cuatro módulos y se evaluarán según el detalle siguiente:

Variable	Ponderaciones	Unidades didácticas
Evaluación de Conocimiento	30%	El ciclo académico comprende 4 módulos.
Evaluación de Producto	35%	
Evaluación de Desempeño	35%	

Siendo el promedio final (PF), el promedio simple de los promedios ponderados de cada módulo (PM1, PM2, PM3 y PM4), calculado de la siguiente manera:

$$PF = (PM1+PM2+PM3+PM4)/4$$

El carácter cuantitativo vigesimal consiste en que la escala valorativa es de cero (0) a veinte (20), para todo proceso de evaluación, siendo once (11), la nota aprobatoria mínima, solo en el caso de determinación de la nota la fracción de 0.5 o más va a favor de la unidad entera inmediata superior (art. 130). Para los currículos de estudio de competencia no se considera el examen sustitutorio (art. 138).

### VIII. BIBLIOGRAFÍA Y REFERENCIAS WEB

1. Joyanes Aguilar, Luis. Lenguajes de programación y estructura de datos. 2002 España Edit. McGraw Hill
2. Flores Cueto, J. y Bertolotti Z., Carmen. Método de las 6'D. 2008 Perú Edit. Universidad San Martín de Porres
3. Pantigoso Silva, R. Jaime. Fundamentos de Programación: Algoritmos y Diagramas de Flujo. 2005 Perú Grupo Editorial Megabyte
4. Weitzenfeld, Alfredo. Ingeniería de Software Orientada a Objetos con UML, Java e Internet 2000 México Internacional Thomson Editores, S.A
5. P.J. Deitel y H.M. Deitel. Java Cómo programar 2008 México Edit. Pearson Educación
6. Bruce Eckel. Piensa en Java. 2003 España Edit. Prentice Hall
7. Óscar Belmonte y otros. Desarrollo de Proyectos Informáticos con Tecnología Java. Bajo licencia de Creative Commons

## IX. PROBLEMAS QUE EL ESTUDIANTE RESOLVERÁ AL FINALIZAR EL CURSO

Magnitud causal objeto del problema	Acción métrica de vinculación	Consecuencia métrica vinculante de la acción
Deficiente conocimiento de la programación orientada a objetos.	<ul style="list-style-type: none"><li>• Afianza sus conocimientos de POO y los aplica en la construcción de programas.</li><li>• Crea sistemas informáticos aplicando la sobrecarga de operadores y métodos.</li></ul>	Se espera que un porcentaje superior al 75% de los estudiantes creen programas utilizando las nociones de la POO tales como clases, objetos, constructores, sobrecarga, etc.
Desconocimiento de herencia y clases abstractas.	<ul style="list-style-type: none"><li>• Reconoce la diferencia entre composición y herencia.</li><li>• Crea programas utilizando clases y métodos abstractos.</li><li>• Crea programas utilizando herencia múltiple mediante el uso de interfaces.</li></ul>	Se espera que un porcentaje superior al 75% de los estudiantes sean capaces de crear programas haciendo uso del concepto de herencia simple y múltiple.
Manejo nulo de conexión a base de datos.	<ul style="list-style-type: none"><li>• Conoce las nociones fundamentales de base de datos.</li><li>• Comprende el uso de las clases de Java para conexión a base de datos y hace una conexión con una base en Access.</li></ul>	Se espera que un porcentaje superior al 75% de los estudiantes comprenda el proceso de conexión a base de datos.
Desconocimiento de los patrones de diseño.	<ul style="list-style-type: none"><li>• Comprende la importancia de los patrones de diseño en la construcción de sistemas informáticos.</li><li>• Reconoce la diferencia entre los diferentes patrones de diseño.</li><li>• Construye programas utilizando patrones de diseño.</li></ul>	Se espera que un porcentaje superior al 75% de los estudiantes construya estructuras utilizando los diferentes patrones de diseño.

Huacho, Abril de 2018.

---

**Lic. Flor E. Ramírez Mundaca**  
**Docente responsable**  
**COMAP N° 1343**