



**UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN**

**Facultad de Ciencias
Escuela Profesional de Física**

SÍLABO POR COMPETENCIAS

CURSO: LENGUAJE DE PROGRAMACIÓN

DOCENTE: Lic. Rosales Morales Julio Martín

SÍLABO DE LENGUAJE DE PROGRAMACIÓN

I. DATOS GENERALES

Línea de carrera	Computación e informática
Curso	Lenguaje de Programación
Código	204
Horas	05 HT: 01; HP: 04
Ciclo	III

II. SUMILLA Y DESCRIPCIÓN DEL CURSO

Sumilla

El curso comprende el estudio y desarrollo de los siguientes temas: Introducción al Sistema Operativo. Algunas órdenes importantes. Introducción al Lenguaje de Programación. Instrucciones de control: condicionales y repetitivas. Datos de tipos definidos. Arreglos o matrices. Sub programas. Funciones. Archivos y registros.

Descripción del curso

La asignatura de Lenguaje de Programación I es de naturaleza teórica - práctica, se orienta al estudio de los principales reglas y nociones de la programación orientada a objetos que permitirán la de creación de clases; con el propósito de 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	Semanas
I	Dada la importancia de la modelación para resolver problemas, formula algoritmos usando las nociones de programación y software informáticos, con precisión.	Conceptos básicos de programación.	4
II	Ante la necesidad de resolver problemas estructurados, aplica los fundamentos de la programación orientada a objetos en la creación de programas para lograr tal fin, con precisión.	El entorno Java. Estructuras condicionales.	4
III	Ante el aumento de la complejidad de los problemas, utiliza cuadros de diálogo y las estructuras de control, basándose en bibliografía especializada.	Cuadros de diálogo. Estructuras repetitivas.	4
IV	En el marco de la modelación matemática, utiliza los arreglos unidimensionales y bidimensionales, las excepciones e interfaces gráficas, consultando diversas fuentes de información confiables.	Estructuras de salto y almacenamiento. Uso de excepciones.	4

IV. INDICADORES DE CAPACIDADES AL FINALIZAR EL CURSO

N°	INDICADORES DE CAPACIDAD AL FINALIZAR EL CURSO
01	Explica los conceptos básicos acerca de la programación y reconoce la diferencia entre los tipos de programación.
02	Aplica los fundamentos de programación en el diseño de algoritmos que representan situaciones problemáticas
03	Codifica algoritmos en pseudocódigo y los prueba con la ayuda de un software como es el PSEInt.
04	Codifica algoritmos en diagramas de flujo y los prueba con la ayuda de un software como es el DFD.
05	Describe la programación orientada a objetos y las características de java.
06	Comprende el uso de datos primitivos, de los operadores matemáticos y relacionales, de las clases Scanner, String y Math.
07	Comprende el uso de las estructuras condicionales if simple, if else, else if e if anidados.
08	Comprende el uso de la estructura condicional de selección múltiple switch case.
09	Construye interfaces gráficas sencillas haciendo uso de la clase JOptionPane con diferentes cuadros de diálogo.
10	Usa las estructuras repetitivas do while y while, estableciendo sus similitudes y diferencias.
11	Comprende el uso de la estructura repetitiva for.
12	Desarrolla programas usando las estructuras antes estudiadas de manera combinada.
13	Comprende el uso de las sentencias break y continue.
14	Define y crea arrays unidimensionales y bidimensionales.
15	Maneja excepciones para el tratamiento de errores en tiempo de ejecución.
16	Crea interfaces gráficas de usuario sencillas haciendo uso de las clases AWT y Swing.

V. DESARROLLO DE LAS UNIDADES DIDÁCTICAS

Capacidad de la unidad didáctica I: Dada la importancia de la modelación para resolver problemas, formula algoritmos usando las nociones de programación y software informáticos, con precisión.

UNIDAD I: Conceptos básicos de programación.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
1	1. Conceptos básicos. Datos de un lenguaje de programación. 2. Programación estructurada y modular. 3. Modelamiento de problemas.	<ul style="list-style-type: none"> Identifica las diferencias entre programación estructurada y modular. Diseña algoritmos. 	<ul style="list-style-type: none"> Participa activamente en clase. Desarrolla un espíritu crítico y constructivo. 	<ul style="list-style-type: none"> Exposición temática con la participación activa de los alumnos. Uso de la tecnología informática. Entrega de trabajos y retroalimentación. Intervenciones orales y debate. Trabajo individual. Trabajo en equipo. 	Explica los fundamentos de programación y las diferencias entre los tipos de programación.	
2	1. Manual de uso de PSeInt. 2. Uso de PSeInt en el modelamiento de programas sencillos con pseudocódigos.	<ul style="list-style-type: none"> Desarrolla algoritmos en pseudocódigo. Manipula el software PSeInt. 	Muestra interés para el aprendizaje y auto gestiona su aprendizaje.		Codifica algoritmos en pseudocódigo y los prueba con PSeInt.	
3	1. Manual de uso de DFD. 2. Uso de DFD en el modelamiento de programas sencillos con diagramas de flujo.	<ul style="list-style-type: none"> Desarrolla algoritmos en diagramas de flujo. Manipula el software DFD para probar diagramas de flujo. 	Reflexiona sobre la importancia de los temas realizando preguntas y buscando información.		Codifica algoritmos en diagramas de flujo y los prueba con DFD.	
4	Evaluación de la unidad:	Conocimientos: Examen práctico.	Producto: Trabajo práctico.	Desempeño: Práctica de cómputo.		

Capacidad de la unidad didáctica II: Ante la necesidad de resolver problemas estructurados, aplica los fundamentos de la programación orientada a objetos en la creación de programas para lograr tal fin, con precisión.

UNIDAD II: El entorno Java. Estructuras condicionales.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
5	1. Fundamentos de la POO. El entorno Java. Estructura de un programa en Java. Escritura y lectura de datos. 2. Asignación. Uso de variables, tipos de datos. La clase String.	<ul style="list-style-type: none"> Construye programas sencillos. Analiza y soluciona problemas por medio de la programación. Desarrolla en forma grupal las preguntas propuestas. 	<ul style="list-style-type: none"> Valora la utilidad de la POO. Colabora en el trabajo y desarrollo de ejercicios en aula. 	<ul style="list-style-type: none"> Exposición temática con la participación activa de los alumnos. Uso de la tecnología informática. Entrega de trabajos y retroalimentación. Intervenciones orales y debate. Trabajo individual. Trabajo en equipo. 	<ul style="list-style-type: none"> Describe las características de POO y de Java. Comprende el uso de tipos de datos y de las clases Scanner y String. 	
6	1. Uso Operadores aritméticos, relacionales y lógicos. La clase Math. 2. Estructuras de control condicionales: sentencia if.	<ul style="list-style-type: none"> Empieza operadores y la clase Math y efectúa operaciones matemáticas. Empieza la estructura if else en problemas con dos condiciones. 	Acepta ideas de los demás y expone sus propias ideas.		<ul style="list-style-type: none"> Usa operadores y la clase Math. Comprende el uso de la estructura if. 	
7	1. Estructuras de control condicionales: sentencia if anidados. 2. Sentencia switch case.	<ul style="list-style-type: none"> Empieza if anidados en problemas con varias condiciones. Reemplaza if anidados por switch. 	Asume el trabajo con responsabilidad y espíritu crítico.		<ul style="list-style-type: none"> Usa if anidados. Comprende el uso de switch case. 	
8	Evaluación de la unidad:	Conocimientos: Examen práctico.	Producto: Guía de Problemas.	Desempeño: Intervenciones orales.		

Capacidad de la unidad didáctica III: Ante el aumento de la complejidad de los problemas, utiliza cuadros de diálogo y las estructuras de control, basándose en bibliografía especializada.

UNIDAD III: Cuadros de diálogo. Estructuras repetitivas.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
9	1. Uso de JOptionPane. 1.1. Cuadros de mensaje. 1.2. Cuadros de entrada de datos. 1.3. Cuadros de confirmación.	<ul style="list-style-type: none"> Ejemplifica y construye diversas interfaces con cuadros de diálogo. Identifica las diferencias entre los diferentes cuadro de diálogo. 	Demuestra buena disposición para el aprendizaje y el trabajo en aula.	<ul style="list-style-type: none"> Exposición temática con la participación activa delos alumnos. Uso de la tecnología informática. Entrega de trabajos y retroalimentación. Intervenciones orales y debate. Trabajo individual. Trabajo en equipo. 	Construye interfaces gráficas sencillas haciendo uso de la clase JOptionPane. Comprende el uso de las estructuras do while y while, y sus diferencias. <ul style="list-style-type: none"> Usa la estructura for. Crea programas combinando las estructuras antes estudiadas. 	
10	1. Estructuras de control repetitivas. 1.1. Sentencia do while. 1.2. Sentencia while.	<ul style="list-style-type: none"> Compara las estructuras de control do while y while. Usa las estructuras do while y while. 	Participa en clase exponiendo y debatiendo sus ideas.			
11	1. Estructuras de control repetitivas. 1.1. Sentencia for. 1.2. Ejercicios combinados.	<ul style="list-style-type: none"> Diseña programas aplicando la estructura for. Emplea las estructuras condicionales y/o repetitivas combinándolos. 	<ul style="list-style-type: none"> Muestra superación. Acepta ideas de los demás y hace un análisis crítico. 			
12	Evaluación de la unidad:	Conocimientos: Práctica dirigida.	Producto: Trabajo de investigación.	Desempeño: Debate.		

Capacidad de la unidad didáctica IV: En el marco de la modelación matemática, utiliza los arreglos unidimensionales y bidimensionales, las excepciones e interfaces gráficas, consultando diversas fuentes de información confiables.

UNIDAD IV: Estructuras de salto y almacenamiento. Uso de excepciones.	Semana	Contenidos			Estrategias Didácticas	Indicadores de logro
		Conceptual	Procedimental	Actitudinal		
13	1. Uso de las sentencias break y continue. 2. Arreglos unidimensionales. 3. Arreglos bidimensionales. 4. Uso de la clase Array.	<ul style="list-style-type: none"> Usa las sentencias de salto break y continue. Almacena información usando arrays y sus métodos. 	<ul style="list-style-type: none"> Asume el trabajo con buena actitud. Colabora con sus demás compañeros. 	<ul style="list-style-type: none"> Exposición temática con la participación activa delos alumnos. Uso de la tecnología informática. Entrega de trabajos y retroalimentación. Intervenciones orales y debate. Trabajo individual. Trabajo en equipo. 	<ul style="list-style-type: none"> Comprende el uso de break y continue. Define y crea arrays y usa sus métodos. Maneja los distintos tipos de excepciones para el tratamiento de errores en tiempo de ejecución. Crea interfaces gráficas de haciendo uso de las clases AWT y Swing.	
14	1. Introducción a las excepciones: try y catch. 2. Manejo de excepciones: throws, throw y finally.	<ul style="list-style-type: none"> Maneja y trata errores por medio de excepciones. Identifica las diferencias entre los diferentes tipos de excepciones. 	Fomenta un ambiente de compañerismo para discutir y analizar los problemas propuestos.			
15	1. Uso de AWT y Swing: creación de interfaces gráficas de usuario sencillas. Uso de formularios.	Diseña interfaces gráficas sencillas y establece las diferencias entre las clases AWT y swing.	Discute sobre la importancia del tema estudiado.			
16	Evaluación de la unidad:	Conocimientos: Evaluación escrita.	Producto: Proyecto de software.	Desempeño: 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 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
Desconocimiento de los conceptos y estructuras básicas utilizadas en programación.	<ul style="list-style-type: none">• Crea algoritmos para solucionar problemas sencillos de programación.• Utiliza las estructuras de control para crear algoritmos.• Maneja software especializado en pseudocódigo y diagramas de flujo.	Se espera que un porcentaje superior al 75% de los estudiantes planteen algoritmos de solución de diversos problemas usando pseudocódigos y diagramas de flujo y software especializados.
Desconocimiento de un lenguaje de programación de distribución libre como es el Java basado en la plataforma Netbeans.	<ul style="list-style-type: none">• Conoce los fundamentos de la programación orientada a objetos.• Crea programas sencillos utilizando el lenguaje de programación Java.• Maneja las estructuras de selección.	Se espera que un porcentaje superior al 75% de los estudiantes sean capaces de crear programas para solucionar situaciones problemáticas y manejar las estructuras de selección eficientemente.
Manejo nulo de cuadros de diálogo y estructuras de repetición.	<ul style="list-style-type: none">• Crea programas utilizando diversos cuadros de diálogos.• Maneja las estructuras de repetición y comprende la diferencia entre ellas.	Se espera que un porcentaje superior al 75% de los estudiantes usen correctamente cuadros de diálogo y las estructuras de repetición.
Manejo nulo de las estructuras de salto y almacenamiento, de excepciones y de interfaces gráficas sencillas.	<ul style="list-style-type: none">• Maneja eficientemente las estructuras de salto.• Usa arreglos y métodos de la clase Array.• Comprende el uso de excepciones para el control de errores.• Crea interfaces gráficas sencillas usando la clase Swing.	Se espera que un porcentaje superior al 75% de los estudiantes creen interfaces gráficas utilizando las clases y estructuras estudiadas hasta ahora.

Huacho, abril de 2018.

Lic. Rosales Morales Julio Martín
Docente Responsable
COESPE N° 1083