



FACULTAD DE CIENCIAS

ESCUELA PROFESIONAL DE ESTADÍSTICA E INFORMÁTICA

MODALIDAD NO PRESENCIAL

SÍLABO POR COMPETENCIAS

CURSO:

LENGUAJE DE PROGRAMACIÓN II

I. DATOS GENERALES

Línea de carrera	Analista de sistemas informáticos
Semestre académico	2020 - I
Código del curso	201
Créditos	4
Horas semanales	Horas totales: 06 Teóricas: 02; Prácticas: 04
Ciclo	III
Sección	A
Apellidos y nombres del docente	Flor Eonice Ramírez Mundaca
Correo institucional	framirez@unifsc.edu.pe
N° de celular	927 630 604

II. 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.

III. CAPACIDADES AL FINALIZAR EL CURSO

	Capacidad de la unidad didáctica	Nombre de la unidad didáctica	Semanas
Unidad I	Al iniciar el aprendizaje de los fundamentos básicos de la programación orientada a objetos, crea programas sencillos, en base a un análisis bibliográfico.	Conceptos básicos de la POO. Sobrecarga.	1 - 4
Unidad II	Ante la necesidad de resolver situaciones más complejas, aplica los principios de herencia, polimorfismo, clases abstractas en la creación de programas, con precisión.	Clases derivadas y polimorfismo. Clases abstractas e interfaces.	5 - 8
Unidad 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.	Interfaces gráficas de usuario. Acceso a base de datos.	9 - 12
Unidad IV	A fin de resolver problemas de diseño de las aplicaciones informáticas, describe los principales patrones de diseño usados en la POO, consultando diversas fuentes de información confiables.	Patrones de diseño. El patrón Modelo/Vista/Controlador (M/V/C).	13 - 16

IV. INDICADORES DE CAPACIDADES AL FINALIZAR EL CURSO

N°	INDICADORES DE CAPACIDAD AL FINALIZAR EL CURSO
01	Emplea los conceptos básicos de la programación orientada a objetos con acierto.
02	Aplica los fundamentos de la POO en el diseño de programas sencillos con coherencia.
03	Utiliza adecuadamente los métodos y constructores en el diseño de proyectos de programación.
04	Ejemplifica con precisión la sobrecarga de operadores y métodos mediante aplicaciones de programación.
05	Discrimina el concepto de composición del concepto de herencia de clases coherentemente.
06	Indica cómo funciona el polimorfismo de clases de manera rigurosa.
07	Describe el uso de las clases y métodos abstractos de manera correcta.
08	Sustenta las ventajas de usar interfaces simples y anidadas de manera ordenada.
09	Usa adecuadamente las diferentes clases del paquete estándar como String, Date, Calendar, etc.
10	Manipula con precisión clases recubridoras para referenciar datos primitivos usados en colecciones.
11	Crea correctamente 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	Analiza la importancia de utilizar los principios de la POO y los patrones con minuciosidad.
14	Fundamenta con pertinencia la importancia del uso del patrón Modelo/Vista/Controlador en diversas aplicaciones informáticas y la adaptación del modelo para aplicaciones web.
15	Emplea cuidadosamente 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 cuidadosamente 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

UNIDAD I. Conceptos básicos de la POO. Sobrecarga.	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.						
	Sem.	Contenidos			Estrategias de la enseñanza virtual	Indicadores de logro de la capacidad	
		Conceptual	Procedimental	Actitudinal			
	1	1. Conceptos básicos de POO: clases, objetos, encapsulamiento. 2. Nociones. Coherencia.	<ul style="list-style-type: none"> • Discute los fundamentos de la POO. • Crea proyectos informáticos sencillos. 	<ul style="list-style-type: none"> • Participa activamente en clase. • Desarrolla un espíritu crítico y constructivo. 	Expositiva (Docente/alumno) <ul style="list-style-type: none"> • Uso de Google Meet. Debate dirigido (Discusiones) <ul style="list-style-type: none"> • Foros, chat. Lecturas <ul style="list-style-type: none"> • Uso de repositorios digitales Lluvia de ideas (Saberes previos) <ul style="list-style-type: none"> • Foros, chat. 	<ul style="list-style-type: none"> • Emplea los conceptos básicos de la POO con acierto. • Aplica los fundamentos de POO en programas sencillos con coherencia. 	
	2	1. Definición de métodos, tipos. 2. Constructores.	<ul style="list-style-type: none"> • Introduce el uso de métodos para codificar programas. • Emplea los constructores. 	Muestra interés para el aprendizaje y auto gestiona su aprendizaje.			Utiliza adecuadamente los métodos y constructores en el diseño de proyectos de programación.
	3	1. Sobrecarga de operadores.	<ul style="list-style-type: none"> • Formula algoritmos usando la sobrecarga de operadores. 	Reflexiona sobre la importancia de los temas realizando preguntas y buscando información.			Ejemplifica con precisión la sobrecarga de operadores y métodos mediante aplicaciones de programación.
	4	2. Sobrecarga de métodos y constructores.	<ul style="list-style-type: none"> • Crea programas usando la sobrecarga de métodos. 				
	EVALUACIÓN DE LA UNIDAD DIDÁCTICA						
	Evidencia de conocimientos			Evidencia de producto		Evidencia de desempeño	
	<ul style="list-style-type: none"> • Estudios de casos • Cuestionarios y exámenes virtuales 			<ul style="list-style-type: none"> • Trabajos individuales y/o grupales • Soluciones a ejercicios propuestos 		<ul style="list-style-type: none"> • Comportamiento en clase virtual y chat • Exposiciones y discusiones 	

UNIDAD II. Clases derivadas y polimorfismo. Clases abstractas e interfaces.	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.						
	Sem.	Contenidos			Estrategias de la enseñanza virtual	Indicadores de logro de la capacidad	
		Conceptual	Procedimental	Actitudinal			
	5 6	1. Composición. 2. Extensión de clases: concepto de Herencia. 3. Polimorfismo.	<ul style="list-style-type: none"> • Emplea la composición y herencia para crear proyectos de software. • Construye programas usando polimorfismo. 	<ul style="list-style-type: none"> • Valora la utilidad de la POO. • Colabora en el trabajo y desarrollo de ejercicios en clase. 	Expositiva (Docente/alumno) <ul style="list-style-type: none"> • Uso de Google Meet. Debate dirigido (Discusiones) <ul style="list-style-type: none"> • Foros, chat. Lecturas <ul style="list-style-type: none"> • Uso de repositorios digitales Lluvia de ideas (Saberes previos) <ul style="list-style-type: none"> • Foros, chat. 	<ul style="list-style-type: none"> • Discrimina la composición de la herencia de clases. • Indica cómo funciona el polimorfismo de clases. 	
	7	1. Definición de clases abstractas y métodos abstractos.	<ul style="list-style-type: none"> • Desarrolla programas utilizando el concepto de clase abstracta. • Formula métodos abstractos. 	Acepta ideas de los demás y expone sus propias ideas.			Describe el uso de las clases y métodos abstractos correctamente.
	8	1. Interfaces. 2. Clases e interfaces anidados.	<ul style="list-style-type: none"> • Compara la eficacia de utilizar una clase abstracta o una interfaz. • Desarrolla las preguntas propuestas. 	Asume el trabajo con responsabilidad y espíritu crítico.			<ul style="list-style-type: none"> • Sustenta las ventajas de usar interfaces ordenada. • Usa interfaces anidadas.
	EVALUACIÓN DE LA UNIDAD DIDÁCTICA						
	Evidencia de conocimientos			Evidencia de producto		Evidencia de desempeño	
	<ul style="list-style-type: none"> • Estudios de casos • Cuestionarios y exámenes virtuales 			<ul style="list-style-type: none"> • Trabajos individuales y/o grupales • Soluciones a ejercicios propuestos 		<ul style="list-style-type: none"> • Comportamiento en clase virtual y chat • Exposiciones y discusiones 	

UNIDAD III. Interfaces gráficas de usuario. Acceso a base de datos.	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.						
	Sem.	Contenidos			Estrategias de la enseñanza virtual	Indicadores de logro de la capacidad	
		Conceptual	Procedimental	Actitudinal			
	9	1. Clases del paquete estándar. 1.1. Cadenas. Recubridoras. 1.2. Colecciones. 1.3. La clase Date, Calendar.	<ul style="list-style-type: none"> • Discute la utilidad de usar la clase Sting Date, Calendar, etc. • Identifica qué es una colección. 	Demuestra buena disposición para el aprendizaje y el trabajo en clase.	Expositiva (Docente/alumno) <ul style="list-style-type: none"> • Uso de Google Meet. Debate dirigido (Discusiones) <ul style="list-style-type: none"> • Foros, chat. Lecturas <ul style="list-style-type: none"> • Uso de repositorios digitales Lluvia de ideas (Saberes previos) <ul style="list-style-type: none"> • Foros, chat. 	<ul style="list-style-type: none"> • Usa clases paquete estándar: String, Date, Calendar. • Manipula clases recubridoras y las colecciones. 	
	10	1. Interfaces gráficas de usuario. 1.1. Contenedor. Componente 1.2. Eventos.	<ul style="list-style-type: none"> • Identifica un contenedor de un componente. • Codifica diversos eventos. 	Participa en clase exponiendo y debatiendo sus ideas.			<ul style="list-style-type: none"> • Crea correctamente interfaces gráficas de usuario 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"> • Construye aplicaciones con acceso a base de datos. • Utiliza las diferentes sentencias SQL. 	<ul style="list-style-type: none"> • Muestra superación. • Acepta ideas de los demás y hace un análisis crítico. 			<ul style="list-style-type: none"> • Emplea con eficacia los elementos necesarios que permita la gestión de una base de datos en Excel.
	12	EVALUACIÓN DE LA UNIDAD DIDÁCTICA					
	Evidencia de conocimientos		Evidencia de producto		Evidencia de desempeño		
	<ul style="list-style-type: none"> • Estudios de casos • Cuestionarios y exámenes virtuales 		<ul style="list-style-type: none"> • Trabajos individuales y/o grupales • Soluciones a ejercicios propuestos 		<ul style="list-style-type: none"> • Comportamiento en clase virtual y chat • Exposiciones y discusiones 		

UNIDAD IV. Patrones de diseño. El patrón Modelo/Vista/Controlador MVC.	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.						
	Sem.	Contenidos			Estrategias de la enseñanza virtual	Indicadores de logro de la capacidad	
		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"> • Discute los principios de la POO. • Discute la importancia del uso de los distintos patrones de diseño tal como M/V/C. 	<ul style="list-style-type: none"> • Asume el trabajo con buena actitud. • Colabora con sus demás compañeros. 	Expositiva (Docente/alumno) <ul style="list-style-type: none"> • Uso de Google Meet. Debate dirigido (Discusiones) <ul style="list-style-type: none"> • Foros, chat. Lecturas <ul style="list-style-type: none"> • Uso de repositorios digitales Lluvia de ideas (Saberes previos) <ul style="list-style-type: none"> • Foros, chat. 	<ul style="list-style-type: none"> • Analiza la importancia de los principios de la POO. • Fundamenta la importancia del 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.			<ul style="list-style-type: none"> • Emplea los patrones de diseño <i>Singleton</i>, <i>Factory Method</i> y <i>Abstract Factory</i>.
	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 de los temas estudiados.			<ul style="list-style-type: none"> • Emplea los patrones de diseño <i>Strategy</i>, <i>Observer</i> y <i>Decorator</i>; coherentemente.
	16	EVALUACIÓN DE LA UNIDAD DIDÁCTICA					
	Evidencia de conocimientos		Evidencia de producto		Evidencia de desempeño		
	<ul style="list-style-type: none"> • Estudios de casos • Cuestionarios y exámenes virtuales 		<ul style="list-style-type: none"> • Trabajos individuales y/o grupales • Soluciones a ejercicios propuestos 		<ul style="list-style-type: none"> • Comportamiento en clase virtual y chat • Exposiciones y discusiones 		

VI. MATERIALES EDUCATIVOS Y OTROS RECURSOS DIDÁCTICOS

Se utilizarán todos los materiales y recursos requeridos, de acuerdo a la naturaleza de los temas programados. Básicamente serán:

1. MEDIOS Y PLATAFORMAS VIRTUALES	2. MEDIOS INFORMÁTICOS
<ul style="list-style-type: none"> • Casos prácticos • Pizarra interactiva • Google Meet • Repositorios de datos 	<ul style="list-style-type: none"> • Computadoras • Tablet • Celulares • Internet

VII. EVALUACIÓN

La evaluación es inherente al proceso de enseñanza aprendizaje y será continua y permanente. Los criterios de evaluación son de conocimiento, de desempeño y de producto.

1. Evidencias de conocimiento

La evaluación será a través de pruebas escritas y orales para el análisis y autoevaluación. En cuanto al primer caso, medir la competencia a nivel interpretativo, argumentativo y propositivo, para ello debemos ver cómo identifica (describe, ejemplifica, relaciona, reconoce, explica, etc.); y la forma en que argumenta (plantea una afirmación, describe las refutaciones en contra de dicha afirmación, expone sus argumentos contra las refutaciones y llega a conclusiones) y la forma en que propone a través de establecer estrategias, valoraciones, generalizaciones, formulación de hipótesis, respuesta a situaciones, etc. En cuanto a la autoevaluación permite que el estudiante reconozca sus debilidades y fortalezas para corregir o mejorar. Las evaluaciones de este nivel serán de respuestas simples y otras con preguntas abiertas para su argumentación.

2. Evidencias de desempeño

Esta evidencia pone en acción recursos cognitivos, recursos procedimentales y recursos afectivos; todo ello en una integración que evidencia un saber hacer reflexivo; en tanto, se puede verbalizar lo que se hace, fundamentar teóricamente la práctica y evidenciar un pensamiento estratégico, dado en la observación en torno a cómo se actúa en situaciones impredecibles. La evaluación de desempeño se evalúa ponderando como el estudiante se hace investigador aplicando los procedimientos y técnicas en el desarrollo de las clases a través de su asistencia y participación asertiva.

3. Evidencias de producto

Están implicadas en las finalidades de la competencia, por tanto, no es simplemente la entrega del producto, sino que tiene que ver con el campo de acción y los requerimientos del contexto de aplicación. La evaluación del producto se evidencia en la entrega oportuna de sus trabajos parciales y el trabajo final.

Además, se tendrá en cuenta la asistencia como componente del desempeño, el 30% de inasistencia inhabilita el derecho a la evaluación.

Variable	Ponderaciones	Unidades didácticas denominadas módulos
Evaluación de Conocimiento	30%	El ciclo académico comprende 4
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):

$$PF = \frac{PM1 + PM2 + PM3 + PM4}{4}$$

VIII. BIBLIOGRAFÍA

8.1. Fuentes bibliográficas

1. Joyanes, L. (2002). Lenguajes de programación y estructura de datos. España. Edit. McGraw Hill.
2. Flores, J. y Bertolotti, C. (2008). Método de las 6'D. Perú. Edit. Universidad San Martín de Porres.
3. Pantigoso, R. (2005). Fundamentos de Programación: Algoritmos y Diagramas de Flujo. Perú. Grupo Editorial Megabyte.
4. Weitzenfeld, A. (2000). Ingeniería de Software Orientada a Objetos con UML, Java e Internet. México. Internacional Thomson Editores, S.A.
5. P.J. Deitel y H.M. Deitel. (2008). Java Cómo programar. México. Edit. Pearson Educación.
6. Eckel, B. (2003). Piensa en Java. España. Edit. Prentice Hall.
7. Belmonte, O. y otros. (1999) Desarrollo de Proyectos Informáticos con Tecnología Java. Bajo licencia de Creative Commons

8.2. Fuentes electrónicas

1. https://www.etsisi.upm.es/sites/default/files/curso_2013_14/MASTER/MIW.JEE.POOJ.pdf
2. http://www.cua.uam.mx/pdfs/revistas_electronicas/libros-electronicos/2016/2intro-poo/programacion_web.pdf
3. http://www.cartagena99.com/recursos/programacion/apuntes/POOJava_cursolndra.pdf
4. <http://www.it.uc3m.es/java/git-gisc/units/oo-herencia/slides/ProgramacionOrientadaAObjetos.pdf>
5. http://mmc.geofisica.unam.mx/acl/Herramientas/Lenguajes/Java/JavaBasico/Introduccion_POO_coon_Java.pdf

Huacho, Julio de 2020.



Universidad Nacional
"José Faustino Sánchez Carrión"

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