



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN
FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

SÍLABO POR COMPETENCIAS

CURSO: Programación Orientada a Objetos

Docente: Loncán Salazar, Pierre Paul



SÍLABO DE PROGRAMACION ORIENTADA A OBJETOS

I. DATOS GENERALES

LÍNEA DE CARRERA	Sistemas de Información
CURSO	Programación Orientada a Objetos
CÓDIGO	3205154
HORAS	Teoría (3) – Práctica (2)
CICLO	2
SEMESTRE ACADÉMICO	2018-I
PLAN DE ESTUDIOS	5

II. SUMILLA Y DESCRIPCIÓN DEL CURSO

El curso de Programación Orientada a Objetos es de carácter teórico-práctico y tiene el propósito de proporcionar a los estudiantes los conocimientos necesarios para satisfacer los requerimientos de información de las empresas a través de la creación de programas informáticos. El contenido incluye temas relacionados con los principios básicos de la programación orientada a objetos en entornos visuales, las estructuras de almacenamiento estático, el control de errores y la programación en niveles y las bases de datos relacionales y su uso en programas informáticos.

El curso se desarrollará en 16 semanas, teórico-prácticas, es decir 03 horas de teoría y 02 horas de prácticas



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN

FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

III. CAPACIDADES AL FINALIZAR EL CURSO

	CAPACIDAD DE LA UNIDAD DIDACTICA	NOMBRE DE LA UNIDAD DIDACTICA	SEMANAS
UNIDAD I	Analiza el paradigma Orientado a Objetos, mostrando interés por nuevas formas de programación, valorando sus respectivos usos y demostrando un espíritu participativo y crítico	PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS EN ENTORNO VISUAL	1,2,3,4
UNIDAD II	Construye programas utilizando arreglos para el almacenamiento de toda clase de información; comunicándose de forma asertiva en la solución de problemas	ESTRUCTURAS DE ALMACENAMIENTO ESTÁTICO	5,6,7,8
UNIDAD III	Construye programas utilizando manejo de excepciones, polimorfismo, herencias e interfaces; comunicándose de forma asertiva en la solución de problemas	CONTROL DE ERRORES Y PROGRAMACION EN NIVELES	9,10,11,12
UNIDAD IV	Construye programas con acceso a base de datos con responsabilidad y ética en el cumplimiento de sus actividades	BASES DE DATOS RELACIONALES Y SU USO EN PROGRAMAS INFORMÁTICOS	13,14,15,16



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN

FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

IV. INDICADORES DE CAPACIDADES AL FINALIZAR EL CURSO

NÚMERO	INDICADORES DE CAPACIDAD AL FINALIZAR EL CURSO
1	Analiza los programas que habrá de implementar con criterios de Orientación a Objetos: Clases, Objetos, las estructuras de control funciones, procedimientos y aplicaciones visuales a través del Java
2	Aplica controles de flujo condicionales en la construcción de programas
3	Aplica controles de flujo repetitivos en la construcción de programas
4	Aplica el paquete swing en la construcción de programas
5	Aplica la composición y reutilización de objetos en la construcción de programas teniendo en cuenta sus componentes visuales para diseño de programas
6	Aplica los constructores de las clases y su sobrecarga y su utilidad en la construcción de programas.
7	Aplica los array para almacenar información de cualquier tipo en la construcción de programas.
8	Aplica los array de Objetos para almacenar información del tipo registro en la construcción de programas.
9	Aplica el manejo de excepciones para manipular el tratamiento de excepciones en la ejecución de las instrucciones.
10	Aplica la herencia y la reutilización de objetos para disminuir la redundancia de datos.
11	Aplica polimorfismo en el tratamiento de objetos para evitar la declaración de múltiples objetos
12	Analiza las interfaces en la herencia múltiple para evitar múltiples declaraciones de atributos.
13	Aplica las relaciones entre clases para diseñar bases de datos.
14	Aplica las conexiones ODBC para acceder a la información de una base de datos en la construcción de programas.
15	Aplica programas con acceso a Base de datos como soluciones integrales para las organizaciones.
16	Aplica la Programación Orientada a Objetos en la construcción de programas informáticos para la solución de problemas planteados en una organización.



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN

FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

V. DESARROLLO DE LAS UNIDADES DIDACTICAS:

Unidad Didáctica I : OBJETOS EN ENTORNO VISUAL PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN ORIENTADA A	CAPACIDAD DE LA UNIDAD DIDÁCTICA I : Analiza el paradigma Orientado a Objetos, mostrando interés por nuevas formas de programación, valorando sus respectivos usos y demostrando un espíritu participativo y crítico					
	Semana	Contenidos			Estrategia didáctica	Indicadores de logro de la capacidad
		Cognitivos	Procedimental	Actitudinal		
	1	TODO ES UN OBJETO Y CLASES <ul style="list-style-type: none"> ▪ Referencias, tipos primitivos, objetos y clases. ▪ Métodos, parámetros y valores de retorno. ▪ Construcción de un programa: visibilidad. ▪ Clases: atributos y métodos. ▪ Objetos: instancias de clases. 	<ul style="list-style-type: none"> ▪ Reconoce los tipos de Datos ▪ Identifica los elementos de una clase ▪ Crea programas usando clases 	<ul style="list-style-type: none"> ▪ Muestra entusiasmo y una actitud favorable hacia el uso de Clases y Objetos en la programación 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Analiza los programas que habrá de implementar con criterios de Orientación a Objetos: Clases, Objetos, las estructuras de control funciones, procedimientos y aplicaciones visuales a través del Java
	2	CONTROL DEL FLUJO CONDICIONAL DEL PROGRAMA. <ul style="list-style-type: none"> ▪ Declaración con métodos con diferentes algoritmos fundamentales. ▪ Control de ejecución: if-else, switch. 	<ul style="list-style-type: none"> ▪ Identifica los controles de flujo condicional ▪ Crea programas usando controles condicionales 	<ul style="list-style-type: none"> ▪ Respeta las opiniones de otras personas sobre el control de flujos condicionales 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica controles de flujo condicionales en la construcción de programas
	3	CONTROL DEL FLUJO REPETITIVO DEL PROGRAMA. <ul style="list-style-type: none"> ▪ Declaración con métodos con diferentes algoritmos fundamentales. ▪ Control de ejecución: do-while, while, for, break, continue, return. 	<ul style="list-style-type: none"> ▪ Identifica los controles de flujo repetitivos ▪ Crea programas usando controles repetitivos 	<ul style="list-style-type: none"> ▪ Respeta las opiniones de otras personas sobre el control de flujos repetitivos 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica controles de flujo repetitivos en la construcción de programas
	4	APLICACIONES VISUALES <ul style="list-style-type: none"> ▪ Componentes visuales ▪ Desarrollo de aplicaciones 	<ul style="list-style-type: none"> ▪ Identifica los componentes visuales de un programa. ▪ Crea programas usando componentes visuales 	<ul style="list-style-type: none"> ▪ Valora el uso del paquete swing en el desarrollo de las aplicaciones visuales en JAVA. 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica el paquete swing en la construcción de programas
EVALUACIÓN DE LA UNIDAD DIDÁCTICA						
EVIDENCIA DE CONOCIMIENTOS		EVIDENCIA DE PRODUCTO		EVIDENCIA DE DESEMPEÑO		
<ul style="list-style-type: none"> ▪ Examen escrito ▪ Sustentación oral ▪ Exposiciones de los informes presentados 		<ul style="list-style-type: none"> ▪ Informe de la creación de un programa en entorno visual utilizando clases, objetos, controles condicionales y repetitivos. 		<ul style="list-style-type: none"> ▪ Lista de cotejo ▪ Observación en el uso de clases, objetos, controles condicionales y repetitivos, en entornos visuales. 		



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN

FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

Unidad Didáctica II : L ESTRUCTURAS DE ALMACENAMIENTO ESTÁTICO	CAPACIDAD DE LA UNIDAD DIDÁCTICA II : Construye programas utilizando arreglos para el almacenamiento de toda clase de información; comunicándose de forma asertiva en la solución de problemas					
	Semana	Contenidos			Estrategia didáctica	Indicadores de logro de la capacidad
		Cognitivos	Procedimental	Actitudinal		
	5	COMPOSICIÓN: REUTILIZACIÓN. <ul style="list-style-type: none"> Sintaxis de la composición. Referencias a objetos como atributos. Palabra clave: final Datos miembro static 	<ul style="list-style-type: none"> Reconoce los usos de final y static Crea programas usando final y static 	<ul style="list-style-type: none"> Reflexiona sobre la sintaxis de la composición como herramienta de soluciones. 	<ul style="list-style-type: none"> Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> Aplica la composición y reutilización de objetos en la construcción de programas teniendo en cuenta sus componentes visuales para diseño de programas
	6	INICIALIZACION Y LIMPIEZA. <ul style="list-style-type: none"> Constructores y sobrecarga de métodos. Constructores por defecto y con parámetros 	<ul style="list-style-type: none"> Identifica los tipos de constructores Diseña sus propios constructores Crea programas usando distintos tipos de constructores 	<ul style="list-style-type: none"> Toma iniciativa en la búsqueda de soluciones con los temas de inicialización y limpieza. 	<ul style="list-style-type: none"> Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> Aplica los constructores de las clases y su sobrecarga y su utilidad en la construcción de programas.
	7	ESTRUCTURAS ESTÁTICAS <ul style="list-style-type: none"> Tipos de arreglos según el número de sus dimensiones Definición de Arreglos. 	<ul style="list-style-type: none"> Reconoce la estructura de un arreglo Identifica los tipos de arreglos y sus aplicaciones Crea programas usando arreglos 	<ul style="list-style-type: none"> Fomenta el trabajo el equipo para el uso de estructuras estáticas. 	<ul style="list-style-type: none"> Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> Aplica los array para almacenar información de cualquier tipo en la construcción de programas.
	8	ARRAY DE OBJETOS <ul style="list-style-type: none"> Inicialización de miembros y arreglos Arreglos de referencias a objetos. Tipos enumerados 	<ul style="list-style-type: none"> Reconoce la estructura de un arreglo de objetos Crea programas usando arreglos de objetos 	<ul style="list-style-type: none"> Participa comunicándose asertivamente sobre el uso de array de objetos 	<ul style="list-style-type: none"> Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> Aplica los array de Objetos para almacenar información del tipo registro en la construcción de programas.
	EVALUACIÓN DE LA UNIDAD DIDÁCTICA					
	EVIDENCIA DE CONOCIMIENTOS		EVIDENCIA DE PRODUCTO		EVIDENCIA DE DESEMPEÑO	
	<ul style="list-style-type: none"> Examen escrito Sustentación oral Exposiciones de los informes presentados 		<ul style="list-style-type: none"> Informe de la creación de un programa en entorno visual utilizando composición, sobrecarga, estructuras estáticas y arreglos de objetos. 		<ul style="list-style-type: none"> Lista de cotejo Observación en el uso de composición, sobrecarga, estructuras estáticas y arreglos de objetos. 	



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN

FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

Unidad Didáctica III : CONTROL DE ERRORES Y PROGRAMACION EN NIVELES	CAPACIDAD DE LA UNIDAD DIDÁCTICA III : Construye programas utilizando manejo de excepciones, polimorfismo, herencias e interfaces; comunicándose de forma asertiva en la solución de problemas					
	Semana	Contenidos			Estrategia didáctica	Indicadores de logro de la capacidad
		Cognitivos	Procedimental	Actitudinal		
	9	OCULTAR LA IMPLEMENTACIÓN Y MANEJO DE ERRORES <ul style="list-style-type: none"> ▪ La unidad de biblioteca: el paquete (package). ▪ Excepciones básicas: parámetros de las excepciones. ▪ Capturar una excepción. ▪ Crea sus propias excepciones 	<ul style="list-style-type: none"> ▪ Reconoce los tipos de excepciones. ▪ Construye sus propias excepciones ▪ Crea programas usando excepciones 	<ul style="list-style-type: none"> ▪ Demuestra creatividad al crear sus propias excepciones 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica el manejo de excepciones para manipular el tratamiento de excepciones en la ejecución de las instrucciones.
	10	HERENCIA: REUTILIZACIÓN. <ul style="list-style-type: none"> ▪ Sintaxis de la herencia. ▪ Protegido (protected) y conversión hacia arriba. ▪ Instrucción super. ▪ Elección entre composición y herencia. 	<ul style="list-style-type: none"> ▪ Reconoce las características del uso de la herencia ▪ Crea programas usando herencia 	<ul style="list-style-type: none"> ▪ Valora los temas de herencia como herramienta en la búsqueda de soluciones 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica la herencia y la reutilización de objetos para disminuir la redundancia de datos.
	11	POLIMORFISMO. <ul style="list-style-type: none"> ▪ Ligadura en la llamada a los métodos. ▪ Clases y métodos abstractos. ▪ Conversión hacia abajo. ▪ Redefinición frente a sobrecarga 	<ul style="list-style-type: none"> ▪ Identifica los tipos de polimorfismo ▪ Construye sus propias clases y métodos abstractos ▪ Crea programas con polimorfismo 	<ul style="list-style-type: none"> ▪ Cumple oportunamente sus tareas asignadas en la clase. 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica polimorfismo en el tratamiento de objetos para evitar la declaración de múltiples objetos
	12	INTERFACES. <ul style="list-style-type: none"> ▪ “Herencia múltiple” con interfaces: implements. ▪ Extender una interfaz con herencia ▪ Interfaces anidadas. 	<ul style="list-style-type: none"> ▪ Identifica las características de una interface ▪ Construye sus propias interfaces ▪ Crea programas usando interfaces 	<ul style="list-style-type: none"> ▪ Se comunica de forma asertiva sobre el uso de las interfaces 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Analiza las interfaces en la herencia múltiple para evitar múltiples declaraciones de atributos.
	EVALUACIÓN DE LA UNIDAD DIDÁCTICA					
EVIDENCIA DE CONOCIMIENTOS		EVIDENCIA DE PRODUCTO		EVIDENCIA DE DESEMPEÑO		
<ul style="list-style-type: none"> ▪ Examen escrito ▪ Sustentación oral ▪ Exposiciones de los informes presentados 		<ul style="list-style-type: none"> ▪ Informe de la creación de un programa en entorno visual utilizando excepciones, herencia, polimorfismo e interfaces. 		<ul style="list-style-type: none"> ▪ Lista de cotejo ▪ Observación en el uso de excepciones, herencia, polimorfismo e interfaces. 		



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN

FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

Unidad Bases de Datos Relacionales y su uso en programas Didáctica IV: Informáticos	CAPACIDAD DE LA UNIDAD DIDÁCTICA IV: Construye programas con acceso a base de datos con responsabilidad y ética en el cumplimiento de sus actividades					
	Semana	Contenidos			Estrategia didáctica	Indicadores de logro de la capacidad
		Cognitivos	Procedimental	Actitudinal		
	13	INTRODUCCIÓN AL DISEÑO DE BASES DE DATOS <ul style="list-style-type: none"> ▪ Relaciones entre clases. ▪ Navegabilidad de la relación. ▪ Multiplicidad. ▪ Instrucciones SQL fundamentales 	<ul style="list-style-type: none"> ▪ Reconoce los componentes de una base de datos ▪ Construye bases de datos ▪ Ejecuta instrucciones SQL en la base de datos. 	<ul style="list-style-type: none"> ▪ Demuestra tolerancia ante las ideas de sus compañeros sobre las aplicaciones con bases de datos. 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica las relaciones entre clases para diseñar bases de datos.
	14	ACCESO A BASES DE DATOS. <ul style="list-style-type: none"> ▪ Implementación de clases con acceso a Bases de Datos. ▪ Las clases y sus servicios. Pruebas de la conexión. 	<ul style="list-style-type: none"> ▪ Identifica y aplica los códigos para realizar conexiones a Base de Datos. 	<ul style="list-style-type: none"> ▪ Colabora en equipo con el desarrollo de las aplicaciones con base de datos 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica las conexiones para acceder a la información de una base de datos en la construcción de programas.
	15	CONSTRUYE APLICACIONES CON ACCESO A BD. <ul style="list-style-type: none"> ▪ Construcción de programas informáticos orientados a objetos con acceso a base de datos, probando la conectividad. 	<ul style="list-style-type: none"> ▪ Crea programas con acceso a Base de Datos 	<ul style="list-style-type: none"> ▪ Valora la implementación de aplicaciones que usen conexiones a Base de Datos 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica programas con acceso a Base de datos como soluciones integrales para las organizaciones.
	16	IMPLEMENTACIÓN DE APLICACIONES CON ACCESO A BD. <ul style="list-style-type: none"> ▪ Probando la conectividad. ▪ Integrando todos los conceptos. ▪ Despliegue de aplicaciones. 	<ul style="list-style-type: none"> ▪ Crea programas con acceso a Base de Datos y empleando reportes 	<ul style="list-style-type: none"> ▪ Muestra responsabilidad y ética en el cumplimiento de sus tareas 	<ul style="list-style-type: none"> ▪ Clase expositiva y aplicación de un laboratorio. 	<ul style="list-style-type: none"> ▪ Aplica la Programación Orientada a Objetos en la construcción de programas informáticos para la solución de problemas planteados en una organización.
EVALUACIÓN DE LA UNIDAD DIDÁCTICA						
EVIDENCIA DE CONOCIMIENTOS		EVIDENCIA DE PRODUCTO		EVIDENCIA DE DESEMPEÑO		
<ul style="list-style-type: none"> ▪ Examen escrito ▪ Sustentación oral ▪ Exposiciones de los informes presentados 		<ul style="list-style-type: none"> ▪ Informe de la creación de un programa en entorno visual utilizando acceso a base de datos. 		<ul style="list-style-type: none"> ▪ Lista de cotejo ▪ Observación en el uso de acceso a base de datos. 		



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN
FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

VI. MATERIALES EDUCATIVOS Y OTROS RECURSOS DIDÁCTICOS

1.- MEDIOS ESCRITOS.

- Libros
- Revistas

2.- MEDIOS VISUALES Y ELECTRÓNICOS.

- pizarrón
- Videos
- Proyector Multimedia

2.- MEDIOS INFORMÁTICOS.

- Internet

VII. EVALUACIÓN

1.- CALIFICACIÓN.

Sistema de calificación: Escala vigesimal (0-20)

2.- EVALUACIÓN DE LOS RESULTADOS DE LAS UNIDADES DIDÁCTICAS.

Evaluación mensual por cada unidad didáctica: Todas las unidades didácticas serán evaluadas en las tres componentes con un puntaje del 0 al 20, obteniéndose tres (03) notas:

ECn: Evaluación de Conocimientos

WECn: Peso para la evaluación de Conocimiento= 0,30

EPn: Evaluación de Producto

WPCn: Peso para la evaluación de Producto= 0,35

EDn: Evaluación de Desempeño

WECn: Peso para la evaluación de conocimiento= 0,35

PMn: Promedio del Módulo

PMn: Promedio del Módulo, con un decimal sin redondeo.

A las notas anteriores se les aplicarán los pesos indicados en la siguiente tabla:

UNIDA DIDÁCTICA	EVIDENCIA DE CONOCIMIENTOS (30%)	EVIDENCIA DE PRODUCTO (35%)	EVIDENCIA DE DESEMPEÑO (35%)
I	EC ₁	EP ₁	ED ₁
II	EC ₂	EP ₂	ED ₂
III	EC ₃	EP ₃	ED ₃
IV	EC ₄	EP ₄	ED ₄

Promedio del Módulo $PMn = (ECn \times WECn + EPn \times WPCn + EDn \times WEDn)$

Donde el PROMEDIO FINAL: $(PM1 + PM2 + PM3 + PM4)/4$



UNIVERSIDAD NACIONAL
JOSÉ FAUSTINO SÁNCHEZ CARRIÓN
FACULTAD DE INGENIERIA INDUSTRIAL, SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERIA SISTEMAS

VII. BIBLIOGRAFÍA Y REFERENCIAS WEB

UNIDAD DIDACTICA I: PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS EN ENTORNO VISUAL

- Thomas W.(2001). Introducción a la programación orientada a objetos con Java. España: McGraw-Hill Interamericana.
- Vasquez P. (2008). Super Java SE for Windows with NetBeans IDE

UNIDAD DIDACTICA II: ESTRUCTURAS DE ALMACENAMIENTO ESTÁTICO

- Heileman, L. (1998). Estructuras de datos, algoritmos y programación orientada a objetos. Madrid Aires: Mcgraw-Hill Interamericana..
- Joyanes, L. (2001). Programación en Java2. Algoritmos, Estructura de Datos y Programación Orientada a Objetos. España: McGraw-Hill / Interamericana de España S.A.U

UNIDAD DIDACTICA III: CONTROL DE ERRORES Y PROGRAMACION EN NIVELES

- Heileman, L. (1998). Estructuras de datos, algoritmos y programación orientada a objetos. Madrid Aires: Mcgraw-Hill Interamericana.
- Joyanes, L. (2001). Programación en Java2. Algoritmos, Estructura de Datos y Programación Orientada a Objetos. España: McGraw-Hill / Interamericana de España S.A.U

UNIDAD DIDACTICA IV: BASES DE DATOS RELACIONALES Y SU USO EN PROGRAMAS INFORMÁTICOS

- Coronel, E. (2009) Desarrollando soluciones con Java y MySQL. Lima: Macro