

PROGRAMA DE ASIGNATURA**DSY1102****Desarrollo Orientado a Objetos****10****Crédito
s Duoc****4****Crédito
s SCT****Presencial****1. ANTECEDENTES GENERALES**

Horas Totales	Docencia Dirigida Presencial	Horas Pedagógicas	Docencia Dirigida No Presencial	Horas Pedagógicas	Prerrequisitos	Horas Cronológicas de Trabajo Autónomo
90	Laboratorio de PC	60	No Requiere	0	FPY1101	40

**2. DESCRIPCIÓN DE LA ASIGNATURA**

Capacidad/es Institucionales	0
Capacidad/es y Competencia/s del Perfil de Egreso	Desarrollar software seguro y de calidad, analizando el ciclo de vida de éste, según las características del proyecto, las mejores prácticas y estándares de la industria.

RESULTADO DE APRENDIZAJE	INDICADOR DE LOGRO
RA1 Desarrolla software, utilizando un lenguaje de programación con paradigma orientado a objetos, para dar respuesta a requerimientos de clientes, considerando buenas prácticas en la solución de la aplicación.	IL 1.1 Aplica los conceptos asociados a la programación orientada a objetos en una solución de software según las necesidades de la organización.
	IL 1.2 Identifica los atributos, posibles comportamientos y sus modificadores de acceso de una clase para dar solución de software a las necesidades de la organización.
	IL 1.3 Aplica conceptos de orientación a objetos en la configuración de software, de acuerdo con los requerimientos del cliente.
	IL 1.4 Valida el funcionamiento de los objetos desde el programa principal del software, para la detección de problemas de funcionamiento de la aplicación.
	IL 1.5 Implementa colaboración de clases considerando conceptos de orientación a objetos en la programación del software.
RA2 Aplica conceptos avanzados del paradigma orientado a objetos en componentes de software, para dar respuesta a requerimientos de clientes.	IL 2.1 Aplica las sentencias de ciclos, para ser representadas en la solución según el requerimiento del usuario en un caso de negocios.
	IL 2.2 Aplica conceptos de encapsulación avanzada permitiendo cambios internos sin afectar otros componentes del sistema del software.
	IL 2.3 Configura una colección para almacenar información temporal en el programa según lo solicitado por el usuario.
	IL 2.4 Aplica los conceptos de herencia en la programación orientada a objetos que permitan dar solución a un problema planteado un caso de negocios.
	IL 2.5 Utiliza abstracción y polimorfismo de manera efectiva, para permitir la creación de interfaces flexibles y genéricas que puedan adaptarse a diferentes situaciones.
RA3 Organiza proyectos de software, utilizando estructura y/o dependencia de paquetes, para dar orden lógico y cohesión a los componentes desarrollados.	IL 3.1 Utiliza los contenedores y componentes para visualizar la interfaz gráfica dando solución a requerimientos solicitados por el usuario un contexto de negocios.
	IL 3.2 Incorpora principios del patrón de arquitectura Modelo-Vista-Controlador (MVC) a una estructura de directorios coherente en el proyecto de software para una organización eficaz de la información.
	IL3.3 Programa sentencias para la ejecución de consultas con base en los datos de un caso.
	IL3.4 Agrupa componentes en módulos con funcionalidad o propósito común para la facilitación de la navegación y el mantenimiento del código.
	IL3.5 Valida la estructura del proyecto basándose en los estándares de la industria y en las buenas prácticas de desarrollo de softwares.



3.SÍNTESIS DE LA RUTA DE APRENDIZAJE

RESULTADOS DE APRENDIZAJE	INDICADOR/ES LOGRO	EXPERIENCIA DE APRENDIZAJE	ACTIVIDADES Y EVALUACIONES	AMBIENTES DE APRENDIZAJE	N° HORAS
RA1	IL1.1 A IL1.8	EA 1 - Paradigma y programación	Act 1.1 - Paradigma Orientación a Objetos	Laboratorio PC Avanzado	5
			Act 1.2 - Lenguaje de Programación Orientada a Objetos	Laboratorio PC Avanzado	5
			Act 1.3 - Programación y Corrección (POO)	Laboratorio PC Avanzado	10
			Ev For 1 - Clases y atributos I	Laboratorio PC Avanzado	2
			Ev Parcial 1 - Clases y atributos II	Laboratorio PC Avanzado	3
RA2	IL2.1 A IL2.5	EA 2 - Herencia y Colecciones	Act 2.1 - Colecciones	Laboratorio PC Avanzado	10
			Act 2.2 - Herencia	Laboratorio PC Avanzado	10
			Act 2.3 - Sobrecarga, Polimorfismo y herencia	Laboratorio PC Avanzado	10
			Ev For 2 - Herencia y Colecciones	Laboratorio PC Avanzado	2
			Ev Parcial 2 - Herencia, polimorfismo y colecciones	Laboratorio PC Avanzado	3
RA3	IL31 A IL3.6	EA 3 - Interfaz, Estructura y Validación	Act 3.1 - Interfaz gráfica	Laboratorio PC Avanzado	5
			Act 3.2 - Ordenamiento de paquetes	Laboratorio PC Avanzado	10
			Act 3.3 - Estructura, Agrupación y validación	Laboratorio PC Avanzado	5
			Ev For 3 - Interfaz, estructura y agrupación	Laboratorio PC Avanzado	2
			Ev Parcial 3 - Interfaz, estructura y validación	Laboratorio PC Avanzado	3
IL1.1,IL1.2 , IL1.5, IL2.3, IL2.4. IL2.5, IL3.1, IL3.2, y IL3.5		Evaluación Final Transversal	Laboratorio PC Avanzado		5



4. EVALUACIONES

EVALUACIÓN	SITUACIÓN EVALUATIVA	DESCRIPCIÓN	PESO DE LA EVALUACIÓN PARCIAL (%)	PONDERACIÓN FINAL
Ev Parcial 1- Clases y atributos II	Ejecución práctica sin presentación	En esta evaluación el estudiante trabajarán en la resolución de un caso para identificar las clases y atributos, realizando la colaboración entre clases y reglas de negocio sencillas para posteriormente, realizar las instancias correspondientes desde main. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.	30%	60%
Ev Parcial 2 Herencia, polimorfismo y colecciones	Ejecución práctica sin presentación	Esta evaluación consiste en la resolución de un caso donde el estudiante debe aplicar colecciones y herencia. Por otro lado, realizar métodos abstractos, polimorfismos, sobrecarga e implementación de interface, entre otros. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.	40%	
Ev Parcial 3- Interfaz, estructura y validación II	Ejecución práctica sin presentación	En esta evaluación deberá aplicar patrón de diseño, donde agrupen elementos en package que tengan similares características, como referencia es MVC. Finalmente, lograr que interactúe de forma eficiente. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.	30%	
Evaluación Final Transversal	Ejecución práctica sin presentación	La evaluación final transversal, es una ejecución práctica sin presentación que considera la resolución de un caso en un contexto de negocio, intencionada para identificar las clases y los atributos, cómo se relacionan, su jerarquía y colecciones, junto con agrupar sus componentes en módulos con funcionalidad o propósito común para la facilitación de la navegación y el mantenimiento del código. El desarrollo de la evaluación se debe realizar de forma local, sin acceso a internet.		40%

*Nota: El/la docente puede agregar evaluaciones parciales o controles adicionales a las señaladas en este documento.



5. BIBLIOGRAFÍA OBLIGATORIA

Sierra, K. Bates B & Gee, T (2022). Head First Java: A Brain-Friendly Guide O'Reilly Media; 3rd edition.