1. DATOS INFORMATIVOS

DOMINIO:	Hábitat, Infraestructura y Movilidad		
CARRERA:	Ingeniería		
Asignatura/Módulo:	Programación III (Programación Orientada a Objetos)		
Paralelo:	1-2		**N° horas: 120**
Plan de estudios:			**H. aprendizaje en contacto con el docente: 48**
Prerrequisitos:	Programación II		
Periodo académico:	2026-01		**H. aprendizaje autónomo: 24**
			H. aprendizaje práctico-experimental: 48
Docenteo o Co-Docente 1:	**Grado académico y título profesional:**	**Co-Docente 2:**	**Grado académico y título profesional:**
Nelson Salgado Reyes	Doctor en Tecnologías de la Infomración y Comunicación		
Breve reseña de la actividad académica y/o		**Breve reseña de la actividad académica y/o profesional:**	
Docente de la PUCE en la formación de Pregrado y Posgrado.		Gerente de TI para las empresas de TEMENOS S.A, Grupo Provefrut - Nintanga	
Indicación de horario de atención al estudiante:			
Tutoría presencial: Lunes 12:00-13:00		**Teléfono:**	1915
Tutoría virtual:		**Correo electrónico:**	nesalgado@puce.edu.ec

2. DESCRIPCIÓN DE LA ASIGNATURA

La Programación Orientada a Objetos (POO) es un paradigma de programación fundamental para el desarrollo de software moderno. En este paradigma, los programas se organizan en torno a objetos, que encapsulan datos (atributos) y comportamiento (métodos). La POO ofrece una serie de ventajas sobre otros paradigmas de programación, como la modularidad, la reutilización del código, la mantenibilidad y la extensibilidad

3. DESCRIPCIÓN DE COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

COMPETENCIAS TRANSVERSALES	COMPETENCIAS INTERDISCIPLINARES DEL DOMINIO	RESULTADOS DE APRENDIZAJE DE LA CARRERA	RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA
1. Humanista y con proyecto vital.	1.Valorar necesidades del entorno para dar respuestas a problemáticas y desafíos del habitat.	1.Crear sistemas de información mediante metodológicas y tecnológicas que contribuyan al progreso de la sociedad para mejorar la eficiencia en el uso de recursos.	1. Analizar los fundamentos de la programación orientada a objetos y sus principales mecanismos, tales como clases, objetos, encapsulación, herencia, polimorfismo e interfaces, para la correcta estructuración de soluciones de software.
2. Comprometido social, política y ambientalmente.	2.Diseñar productos, procesos o sistemas innovadores y responsables con el medio ambiente.	2.Aplicar tecnologías innovadoras y normas legales para generar soluciones tecnológicas sostenibles que permitan mejorar las condiciones de vida de la comunidad a largo plazo.	2. Diseñar aplicaciones orientadas a objetos, organizando clases y relaciones de manera coherente, aplicando principios de diseño orientado a objetos que favorezcan la modularidad, reutilización y mantenibilidad del software.
3. Crítico y analítico.	3.Aplicar conocimientos de forma pertinente y sostenible en relación con los desafíos del habitat.	3.Aplicar técnicas de modelado de datos que le permitan apoyar a la organización en la definición de estrategias e identificar oportunidades de mejora para la sociedad.	3. Aplicar patrones básicos de diseño orientados a objetos, como apoyo al desarrollo de aplicaciones estructuradas y confiables utilizadas en sistemas y contextos relacionados con la ciberseguridad.

4. EVALUACIÓN DE LOGROS DE APRENDIZAJE

PROYECTO DE NIVEL	RETOS	Resultados de aprendizaje de la asignatura	Pesos	Definición del criterio de evaluación del RdA	Ponderación (en porcentaje sobre 100)	Nivel de logro alcanzado al RdA			
						Alcanzado con excelencia (A) (45-50)	Alcanzado muy bueno (B) (40-44)	Alcanzado bueno (C) (30-39)	Pendiente de alcanzar (D) (29 y menos)
Título: Aplicación Orientada a Objetos para la Gestión segura de accesos en ciberseguridad Descripción: Este proyecto tiene como objetivo desarrollar una aplicación orientada a objetos para la gestión segura de accesos, que permita administrar usuarios, roles, permisos y recursos dentro de un sistema informático. A lo largo del curso, los estudiantes deberán construir el proyecto de manera progresiva por etapas, aplicando los fundamentos de la programación orientada a objetos, los principios de diseño orientado a objetos y el desarrollo de al menos un patrón de diseño, integrando cada uno de los resultados de aprendizaje de la asignatura en un contexto propio de la ciberseguridad.	Reto 1: Análisis y Modelado Inicial Orientado a Objetos. En este primer reto, los estudiantes deberán analizar un problema básico de control de accesos en un sistema informático y modelarlo utilizando los fundamentos de la programación orientada a objetos. El reto consiste en identificar y definir las clases principales del sistema, tales como usuarios, roles, permisos y recursos, estableciendo sus atributos y responsabilidades básicas.	1. Analizar los fundamentos de la programación orientada a objetos y sus principales mecanismos, tales como clases, objetos, encapsulación, herencia, polimorfismo e interfaces, para la correcta estructuración de soluciones de software.	33,33	Criterio 1: Identifique y explique correctamente los fundamentos y mecanismos de la programación orientada a objetos, demostrando comprensión conceptual de clases, objetos, encapsulación, herencia, polimorfismo e interfaces.	50 %	Identifica y explica con precisión y profundidad todos los fundamentos y mecanismos de la POO, utilizando terminología técnica adecuada y demostrando comprensión integral.	Identifica y explica correctamente la mayoría de los fundamentos y mecanismos de la POO, con uso adecuado de terminología técnica.	Identifica y describe los fundamentos y mecanismos básicos de la POO, con explicaciones generales o parcialmente correctas.	Presenta dificultades para identificar o explicar los fundamentos y mecanismos de la POO, evidenciando comprensión insuficiente.
	Reto 2: Diseño del Modelo Orientado a Objetos y Control del Flujo del Sistema. En este segundo reto, los estudiantes deberán diseñar el modelo orientado a objetos del sistema de gestión segura de accesos, organizando las clases identificadas en el reto anterior y definiendo de manera clara sus relaciones y comportamientos.			Criterio 2: Analice situaciones o problemas de software y justifique la selección de mecanismos de programación orientada a objetos adecuados para estructurar soluciones coherentes.	50 %	Analiza problemas de software complejos y justifica de manera sólida y coherente la selección de mecanismos POO, estableciendo relaciones claras entre problema y solución.	Analiza correctamente problemas de software y justifica adecuadamente la selección de mecanismos POO en la mayoría de los casos.	Analiza problemas de software simples y justifica de forma básica la selección de mecanismos POO, con argumentación limitada.	No logra analizar adecuadamente los problemas ni justificar la selección de mecanismos POO.
		2. Diseñar aplicaciones orientadas a objetos, organizando clases y relaciones de manera coherente, y aplicando principios de diseño orientado a objetos que favorecen la modularidad, reutilización y mantenibilidad del software.	33,33	Criterio 1: Diseñe modelos orientados a objetos (clases y relaciones) aplicando principios de diseño que garanticen modularidad, reutilización y mantenibilidad del software.	50 %	Diseña modelos orientados a objetos completos y coherentes, aplicando rigurosamente principios de diseño que aseguran modularidad, reutilización y mantenibilidad.	Diseña modelos orientados a objetos adecuados, aplicando correctamente principios de diseño en la mayoría de los elementos.	Diseña modelos orientados a objetos funcionales, con aplicación básica o parcial de principios de diseño.	Presenta modelos orientados a objetos incompletos o incoherentes, con escasa aplicación de principios de diseño.
	Reto 3: Desarrollo Modular con Métodos y Principios de Diseño OO. En este tercer reto, los estudiantes deberán enfocarse en el desarrollo del sistema aplicando métodos (funciones) para resolver procesos más complejos y combinando estructuras de control (condicionales y bucles) dentro de una arquitectura orientada a objetos. Los estudiantes ampliarán la funcionalidad del proyecto de gestión segura de accesos, incorporando operaciones propias de un contexto de ciberseguridad y, al mismo tiempo, deberán descomponer el problema en tareas más simples mediante la creación de métodos específicos por responsabilidad.			Criterio 2: Desarrolle aplicaciones orientadas a objetos funcionales, evidenciando una correcta organización de clases, uso adecuado de relaciones y aplicación consistente de principios de diseño orientado a objetos.	50 %	Desarrolla aplicaciones orientadas a objetos robustas y funcionales, con organización óptima de clases y relaciones, evidenciando buenas prácticas de programación.	Desarrolla aplicaciones orientadas a objetos correctas y funcionales, con adecuada organización de clases y relaciones.	Desarrolla aplicaciones orientadas a objetos básicas, funcionales, pero con organización limitada de clases y relaciones.	Presenta aplicaciones orientadas a objetos con fallos funcionales o mala organización de clases y relaciones.
		3. Aplicar patrones básicos de diseño orientados a objetos, como apoyo al desarrollo de aplicaciones estructuradas y confiables utilizadas en sistemas y contextos relacionados con la ciberseguridad.	33,34	Criterio 1: Reconozca y explique los patrones básicos de diseño orientados a objetos, describiendo su propósito y su aplicabilidad en el desarrollo de software.	50 %	Reconoce y explica con claridad y profundidad patrones básicos de diseño POO, describiendo correctamente su propósito y contexto de uso.	Reconoce y explica adecuadamente patrones básicos de diseño POO, con descripciones correctas.	Reconoce y describe de forma general patrones básicos de diseño POO, con explicaciones limitadas.	Presenta dificultades para reconocer o explicar patrones básicos de diseño POO.
	Reto 4: Persistencia, Auditoría e Integración Final del Sistema. En este último reto, los estudiantes deberán agregar la funcionalidad para guardar y cargar información del sistema en archivos de texto (o formato equivalente), con el fin de mantener la persistencia de los datos y generar trazabilidad básica, tal como se requiere en contextos de ciberseguridad. Esto permitirá que la aplicación conserve un historial de eventos y que el usuario pueda revisar o continuar la gestión de accesos en ejecuciones posteriores.			Criterio 2: Aplique patrones básicos de diseño orientados a objetos en la construcción de soluciones de software, mejorando la estructura, confiabilidad y calidad de aplicaciones en contextos relacionados con la ciberseguridad.	50 %	Aplica de manera pertinente y efectiva patrones de diseño POO, mejorando significativamente la estructura, calidad y confiabilidad del software desarrollado.	Aplica correctamente patrones de diseño POO, logrando mejoras evidentes en la estructura del software.	Aplica patrones de diseño POO de forma básica, con impacto limitado en la calidad del software.	No logra aplicar adecuadamente patrones de diseño POO en las soluciones desarrolladas.

5. METODOLOGÍA

La asignatura Programación III, se desarrolla mediante una metodología de aprendizaje basado en proyectos y retos, que promueve un aprendizaje activo y práctico. Desde el inicio del curso, los estudiantes trabajan en un proyecto transversal, orientado al desarrollo de un sistema de gestión segura de accesos en un contexto de ciberseguridad, integrando de forma continua la teoría con la práctica.
El curso se organiza en cuatro retos progresivos: tres retos formativos enfocados en el análisis, diseño e implementación modular de una aplicación orientada a objetos, y un reto sumativo final, en el que se integran todas las funcionalidades del sistema, incorporando persistencia, auditoría y pruebas. Esta metodología fortalece la autonomía, la resolución

6. RELACIÓN RESULTADOS DE APRENDIZAJE, EXPERIENCIAS DE APRENDIZAJES Y DIMENSIÓN DEL CONOCIMIENTO

PROYECTO DE NIVEL	RETOS	RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA	Semanas	Experiencias / estrategias Aprendizaje en contacto con el docente.	Horas	Recursos	Escenario	Experiencias / estrategias de aprendizaje práctico-experimental:	Horas	Recursos	Escenario	Experiencias / estrategias de aprendizaje autónomo	Horas	Recursos	Escenario	DIMENSIÓN DEL CONOCIMIENTO (conceptos, hecho, procedimientos o principios)
Título: Aplicación Orientada a Objetos para la Gestión segura de accesos en ciberseguridad Descripción: Este proyecto tiene como objetivo desarrollar una aplicación orientada a objetos para la gestión segura de accesos, que permita administrar usuarios, roles, permisos y recursos dentro de un sistema informático. A lo largo del curso, los estudiantes deberán construir el proyecto de manera progresiva por etapas, aplicando los fundamentos de la programación orientada a objetos, los principios de diseño orientado a objetos y el desarrollo de al menos un patrón de diseño, integrando cada uno de los resultados de aprendizaje de la asignatura en un contexto propio de la ciberseguridad.	Reto 1: Análisis y Modelado Inicial Orientado a Objetos En este primer reto, los estudiantes deberán analizar un problema básico de control de accesos en un sistema informático y modelarlo utilizando los fundamentos de la programación orientada a objetos. El reto consiste en identificar y definir las clases principales del sistema, tales como usuarios, roles, permisos y recursos, estableciendo sus atributos y responsabilidades básicas.	1. Analizar los fundamentos de la programación orientada a objetos y sus principales mecanismos, tales como clases, objetos, encapsulación, herencia, polimorfismo e interfaces, para la correcta estructuración de soluciones de software.	1	Tutorias sincrónicas por videoconferencia.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos mediante el análisis de un caso básico de control de accesos. Práctica: Análisis del problema y reconocimiento de entidades principales del sistema. Ejemplo: Analizar un sistema de control de accesos institucional e identificar usuarios, recursos y roles.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	2	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	**1. Análisis y modelado orientado a objetos** 1.1 Fundamentos de la Programación Orientada a Objetos 1.1.1 Paradigma orientado a objetos y su propósito 1.1.2 Ventajas de la POO en la estructuración de sistemas de software
			1	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Experiencias / estrategias de aprendizaje práctico-experimental Resolución de problemas prácticos a través de la identificación de clases y objetos. Práctica: Definición de clases y objetos a partir del problema planteado. Ejemplo: Identificar las clases Usuario, Rol y Recurso y distinguirlas de sus instancias.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	4	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	1.2 Clases y objetos 1.2.1 Diferencia conceptual entre clase y objeto 1.2.2 Representación de objetos en un sistema de software.
			2	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos enfocada en la definición de atributos y responsabilidades. Práctica: Análisis de atributos y responsabilidades de las clases identificadas. Ejemplo: Determinar atributos y responsabilidades básicas de la clase Usuario.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	2	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	1.3 Atributos y responsabilidades de las clases 1.3.1 Estado de un objeto 1.3.2 Responsabilidad como principio de diseño
			2	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos mediante el encapsulación. Práctica: Identificación de atributos públicos y privados. Ejemplo: Definir qué información del Usuario debe estar protegida mediante encapsulación.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	2	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	1.4 Encapsulación 1.4.1 Ocultamiento de información 1.4.2 Acceso controlado a los datos
	Reto 2: Diseño del Modelo Orientado a Objetos y Control del Flujo del Sistema En este segundo reto, los estudiantes deberán diseñar el modelo orientado a objetos del sistema de gestión segura de accesos, organizando las clases identificadas en el reto anterior y definiendo de manera clara sus relaciones y comportamientos.	2. Diseñar aplicaciones orientadas a objetos, organizando clases y relaciones de manera coherente, aplicando principios de diseño orientado a objetos que favorezcan la modularidad, reutilización y mantenibilidad del software.	3	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos mediante el análisis de herencia y polimorfismo. Práctica: Diseño de jerarquías simples de clases. Ejemplo: Modelar distintos tipos de usuarios con comportamientos comunes y específicos.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	4	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	1.5 Herencia y polimorfismo 1.5.1 Reutilización de código mediante herencia 1.5.2 Polimorfismo como mecanismo de extensibilidad
			3	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos mediante el uso de interfaces. Práctica: Identificación de comportamientos comunes en el sistema. Ejemplo: Definir una interfaz para autenticación de usuarios.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	4	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	**2. Diseño del modelo orientado a objetos** 2.1 Interfaces 2.1.1 Contratos de comportamiento 2.1.2 Desacoplamiento mediante interfaces
			4	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos a través del diseño del modelo OO. Práctica: Elaboración del diagrama de clases del sistema. Ejemplo: Representar clases, atributos y métodos en un diagrama.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	4	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	2.2 Modelado orientado a objetos 2.2.1 Identificación de elementos del modelo 2.2.2 Representación gráfica del diseño
			4	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos mediante el análisis de relaciones entre clases. Práctica: Definición de asociaciones y jerarquías. Ejemplo: Establecer relaciones entre Usuario, Rol y Permiso.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	6	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	2.3 Relaciones entre clases 2.3.1 Asociación 2.3.2 Herencia en el modelo
	Reto 3: Desarrollo Modular con Métodos y Principios de Diseño OO En este tercer reto, los estudiantes deberán enfocarse en el desarrollo del sistema aplicando métodos (funciones) para resolver procesos más complejos y combinando estructuras de control (condicionales y bucles) dentro de una arquitectura orientada a objetos. Los estudiantes ampliarán la funcionalidad del proyecto de gestión segura de accesos, incorporando operaciones propias de un contexto de ciberseguridad y, al mismo tiempo, deberán descomponer el problema en tareas más simples mediante la creación de métodos específicos por responsabilidad		5	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos enfocada en la evaluación del diseño. Práctica: Análisis de cohesión y acoplamiento. Ejemplo: Identificar mejoras al modelo para facilitar mantenimiento.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	5	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	2.4 Principios de diseño orientado a objetos 2.4.1 Cohesión 2.4.2 Acoplamiento
			5	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos mediante la definición de métodos. Práctica: Identificación de operaciones principales del sistema. Ejemplo: Definir métodos para registrar usuarios y asignar roles.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	6	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	**3. Desarrollo modular del sistema** 3.1 Métodos en clases 3.1.1 Métodos como representación del comportamiento 3.1.2. Responsabilidad de los métodos
			6	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos con estructuras de control. Práctica: Uso de condicionales dentro de métodos y Repetición de procesos dentro del sistema. Ejemplo: Validar permisos antes de conceder	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	4	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	3.2 Estructuras de control 3.2.1 Condicionales 3.2.2 Bucles
		3. Aplicar patrones básicos de diseño orientados a objetos, como apoyo al desarrollo de aplicaciones estructuradas y confiables utilizadas en sistemas y contextos relacionados con la ciberseguridad.	6	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos orientada a la modularidad. Práctica: Organización del código por responsabilidades. Ejemplo: Separar autenticación, autorización y gestión de usuarios.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	6	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y almacenamiento en la nube). Biblioteca virtual institucional (PUCE) y repositorios académicos digitales.	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	3.3 Modularidad 3.3.1 Separación de responsabilidades 3.3.2 Mantenibilidad del software
	Reto 4: Persistencia, Auditoría e Integración Final del Sistema		7	Tutorias asincrónicas por mensajes, videos, podcast, comeos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para lenguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos mediante persistencia de datos y eventos en un sistema orientado a objetos. Práctica: Diseño y desarrollo de mecanismos para almacenar y recuperar información del sistema, así como registrar eventos básicos generados por la interacción de los usuarios. Ejemplo: Guardar en archivos de texto la información de usuarios y roles del sistema, y registrar los eventos generados durante el	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guia de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Consulta e investigación, resolución de problemas de programación sobre el tema en estudio	6	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y almacenamiento en la nube). Biblioteca virtual institucional (PUCE) y repositorios académicos digitales.	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	**4. Persistencia, auditoría, patrones e integración** 4.1 Persistencia de datos 4.1.1 Almacenamiento y recuperación de información persistente en aplicaciones orientadas a objetos. 4.2 Persistencia de eventos 4.2.1 Registro, almacenamiento y consulta de las actividades y eventos generados por el sistema.

	En este último reto, los estudiantes deberán agregar la funcionalidad para guardar y cargar información del sistema en archivos de texto (o formato equivalente), con el fin de mantener la persistencia de los datos y generar trazabilidad básica, tal como se requiere en contextos de ciberseguridad. Esto permitirá que la aplicación conserve un historial de eventos y que el usuario pueda revisar o continuar la gestión de accesos en ejecuciones posteriores.		7	Tutorias asincrónicas por mensajes, videos, podcast, correos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para leguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos orientados a la auditoría básica y la trazabilidad de acciones en sistemas de información. Práctica: Análisis y uso de los registros de eventos del sistema para identificar, revisar y validar las acciones realizadas por los usuarios. Ejemplo: Revisar los archivos de eventos generados por el sistema para identificar intentos de acceso, asignaciones de roles y acciones relevantes, relacionándolos con	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guía de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Cosulta e investigación, resolución de problemas de programación sobre el tema en estudio	6	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y almacenamiento en la nube). Biblioteca virtual	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	4.3 Auditoría y trazabilidad 4.3.1 Concepto de auditoría en sistemas informáticos y la importancia de la trazabilidad de eventos. 4.4 Uso de registros para auditoría 4.4.1 Análisis de registros de actividad del sistema e identificación de eventos relevantes para el control y seguimiento.
			8	Tutorias asincrónicas por mensajes, videos, podcast, correos electrónicos, foros, chats.	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para leguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos aplicando patrones de diseño. Práctica: Selección de un patrón básico. Ejemplo: Aplicar un patrón para gestionar el registro de eventos.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guía de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Cosulta e investigación, resolución de problemas de programación sobre el tema en estudio	6	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	4.5 Patrones de diseño 4.5.1 Concepto de patrón 4.5.2 Patrones básicos en sistemas OO
			8	Tutorias sincrónicas por videoconferencia. Trabajo Final	2	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. IDE para leguaje JAVA	Entorno virtual de aprendizaje, weblabs, software especializado	Resolución de problemas prácticos de integración final. Práctica: Integración de persistencia, auditoría y diseño OO. Ejemplo: Registrar accesos y permitir su consulta posterior.	1	Dispositivos electrónicos portátiles o de escritorio. Conexión a internet. Plataforma Educativa. Guía de laboratorio. Weblab y/o software especializado.	Entorno virtual de aprendizaje, weblabs, software especializado	Cosulta e investigación, resolución de problemas de programación sobre el tema en estudio	5	Computador portátil o de escritorio con sistema operativo actualizado. Conexión estable a internet. Plataforma virtual institucional (aula virtual). Entorno de desarrollo integrado (IDE) y compilador/intérprete de un lenguaje de programación orientado a objetos (Java). Herramientas TIC para comunicación y colaboración académica (videoconferencia, foros, repositorios de código y almacenamiento en la nube). Biblioteca virtual institucional (PUCE) y repositorios académicos digitales.	Entorno virtual de aprendizaje, weblabs, software especializado, acceso a la biblioteca virtual	4.6 Integración del sistema 4.6.1 Trazabilidad de eventos 4.6.2 Evaluación global de la solución desarrollada en contexto de ciberseguridad

8. BIBLIOGRAFÍA

a. BÁSICA

Bibliografía (basarse en normas APA)	Código Biblioteca PUCE	Nro. de ejemplares
Hernández Bejarano, M., & Baquero Rey, L. E. (2023). Programación orientada a objetos en Java: Buenas prácticas. Ingeniería de Sistemas. ISBN 978-958-792-463-3	SDO029862	1
PLott, S. F., & Phillips, D. (2021). Python object-oriented programming: Build robust and maintainable object-oriented Python applications and libraries (4th ed.). Packt Publishing. ISBN 9781801077262.	PUCE213927	1
Gervais, L. (s. f.). El diseño orientado a objetos: Introducción a la plataforma Java, los tipos en Java, creación de clases, herencia y polimorfismo, comunicación entre objetos, multithreading, pruebas y reflexión. ISBN 978-2-409-01929-6	ESM416386	1
Oviedo Regino, E. M. (2015). Lógica de programación orientada a objetos. Ecoe Ediciones. ISBN 978-958-771-136-3.	PUCE190398 PUCE198175	1

Bibliografía (basarse en normas APA)	Código Biblioteca PUCE	Nro. de ejemplares
Guardati Buemo, S. (2016). Estructuras de datos básicas: Programación orientada a objetos con Java. Alfaomega	https://bibliotecadigitalpuce.puce.elogim.com/library/publication/estructuras-de-datos-basicas-programacion-orientada-a-objetos-con-java-1744232274	1
Blanco Fernández, Y. (2020). Introducción a la programación orientada a objetos. Andavira Editora	https://bibliotecadigitalpuce.puce.elogim.com/library/publication/introduccion-a-la-programacion-orientada-a-objetos-1744232280	
Guardati Buemo, S. (2007). Estructura de datos orientada a objetos: Algoritmos con C++. Pearson Educación	https://bibliotecadigitalpuce.puce.elogim.com/library/publication/estructura-de-datos-orientada-a-objetos-algoritmos-con-c	

Elaborado por:

Nelson Salgado Reyes

f) Docente

Revisado y Aprobado por:

Damián Nicolalde

f) Coordinador de carrera

Fecha: 2 de febrero de 2026