Universidad CENFOTEC	Universidad CENFOTEC

SOFT-04 Programación Orientada a Objetos

1. Datos generales del curso

Código	**SOFT-04**
Nombre	Programación Orientada a Objetos
Créditos	5
Duración	15 semanas
Distribución de horas por semana	Horas lectivas: 6 (Teoría: 3 y Práctica: 3) Horas de trabajo independiente: 9 Total de horas: 15
Modalidad	Presencial
Naturaleza	Teórico – Práctico
Ubicación	4
Requisitos	SOFT-02 Principios de Programación 2
Correquisitos	No aplica
Nivel	Grado
Profesor	Por definir
Sede	Central

2. Descripción del curso

La programación orientada objetos se define como el estudio de la programación que establece el diseño de aplicaciones de software por medio de clases, atributos de datos, relaciones entre objetos e interfaces de usuario para el desarrollo de programas con una estructura más natural asociada a la formulación y a la solución de problemas en la vida real. La importancia del estudio de la programación orientada a objetos en la formación del profesional en ingeniería del software radica en el uso de un lenguaje de programación bajo el paradigma de objetos y la consulta de datos para crear aplicaciones de computadora de gran utilidad y calidad en el mercado laboral. En el curso “Programación Orientada a Objetos”, los estudiantes emplean la programación orientada a objetos para desarrollar soluciones a problemas a través de la integración del software de computadora y las bases de datos. Las principales temáticas por abordar son relaciones entre objetos, relaciones entre clases, modelado de aplicaciones, interfaces y arquitectura de software y las actividades de aprendizaje son la creación de una aplicación de software, la resolución de prácticas de programación y un portafolio de evidencias. Este curso se relaciona con los siguientes rasgos de perfil profesional: “describir las arquitecturas de software y los procesos de la programación orientada a objetos para la elaboración de aplicaciones de software eficientes y legibles”, “construir aplicaciones de software por medio del Lenguaje de Modelado Unificado (UML), bases de datos relacionales e interfaces de usuarios para asegurar la calidad del software” y “desempeñar una actitud crítica e integral en la solución de problemas, que favorece la toma de decisiones en el proceso del diseño de software”.

3. Cuadro de competencia

Disposición: Profesionalismo			
Competencia	**Resultado de aprendizaje**	**Criterios de desempeño**	**Evidencias**
Desarrollar aplicaciones y programas de software para la solución de problemas de alta complejidad y buena definición, utilizando el paradigma orientado a objetos, el diseño de aplicaciones por capas, un lenguaje de programación y de consulta de bases de datos de manera integral.	Diseña una aplicación de software con el paradigma orientado a objetos, usando relaciones entre objetos y clases, modelados de aplicaciones, bases de datos e interfaces.	- Crea un repositorio de la aplicación en una plataforma de control de versiones en línea para tener una versión inicial del software - Confecciona el diagrama de lenguaje unificado de modelado (UML), incluyendo las clases, métodos y atributos de los objetos - Elabora una estructura básica del código en un lenguaje de alto nivel, que contiene los paquetes de la capa lógica y el diseño de interfaz - Elabora el diagrama de lenguaje unificado final (UML), incluyendo asociación, agregación, composición y herencia	**Aplicación de software** -Aplicación con software: con código.

		- Desarrolla un repositorio de la aplicación para la interfaz gráfica con la arquitectura de interfaz de usuario - Desarrolla un repositorio de la aplicación en la plataforma para la lógica de las clases, incluyendo los paquetes, lógica de negocio y entidades - Elabora la interfaz gráfica de la aplicación, siguiendo la arquitectura vista en el curso - Relaciona la interfaz gráfica de la aplicación con la lógica de las clases y la base de datos usando un patrón de objetos de acceso de datos - Implementa la lectura de credenciales y el servidor de base de datos de un archivo de texto	

		- Verifica la funcionalidad de la aplicación, utilizando como mínimo los dos casos de prueba - Realiza una explicación de la aplicación y contesta de forma correcta las preguntas propuestas por el docente. - Incorpora la retroalimentación que el docente brindó a los diferentes avances.	
	Resuelve prácticas de programación, utilizando el paradigma de orientación de objetos, modelado de aplicaciones y arquitectura de software e interfaces.	- Crea un repositorio en una plataforma de versiones, en el cual sube el código generado en la práctica. - Utiliza un lenguaje de alto nivel para implementar la práctica - Elabora un documento, que contiene el proceso de abstracción, atributos y objetos	**Prácticas de programación** -Documento con el proceso de abstracción, atributos y objetos de la solución -Solución de prácticas de programación.

		de la solución de la práctica - Crea las clases de la solución de la práctica en el repositorio en la plataforma de versiones - Genera una librería de la capa lógica y la importa en la interfaz de usuario - Crea una arquitectura de software, permitiendo el registro, enlistar y mostrar datos. - Elabora una interfaz gráfica (en el caso que sea necesario) de la práctica propuesta, usando una arquitectura lógica - Realiza las prácticas de forma satisfactoria en el tiempo establecido por el docente.	
	Elabora un portafolio de evidencias sobre los conceptos, las estructuras y la	- Utiliza alguna herramienta tecnológica de recopilación de información para el	**Portafolio de evidencias** -Documento con la recopilación de información sobre

	arquitectura de la programación orientada a objetos, integrando la conceptualización y ejemplos de su empleo genérico.	desarrollo del portafolio - Diseña el portafolio con uso de elementos multimedia - Comprende las arquitecturas, métodos y modelo de software basado en programación orientada a objetos - Integra en el portafolio ejemplos de las estructuras y empleo genérico en la programación orientada a objetos - Reconoce las arquitecturas y modelos de la programación orientada a objetos - Presenta el portafolio en el tiempo y momento acordado por el docente	conceptos, estructuras y ejemplos.

4. Contenidos temáticos

Módulo 1. Conceptos del paradigma de programación orientado a objetos

- Abstracción.
- Encapsulamiento.
- Modularidad.

Módulo 2. Relaciones entre objetos

- Dependencia.
- Asociación.
- Agregación.
- Composición.
- Representación de relaciones.

Módulo 3. Relaciones entre clases

- Estructura de clases.
- Herencia y redefinición de métodos.
- Modificadores de acceso.
- Clases abstractas y polimorfismo.
- Ligado: estático y dinámico.
- Conversión de tipos.

Módulo 4. Modelado de aplicaciones de software

- Técnicas de modelado.
- Contrato de operaciones.
- Diagramas de interacción.
- Diagramas de actividad.

Módulo 5. Interfaces, expresiones e hilos

- Concepto, terminología e implementación.
- Comparación con clases abstractas.
- Comparación con herencia.
- Manejo de expresiones e hilos

Módulo 6. Arquitectura de software y excepciones

- Arquitectura de múltiples capas
- Conexión a bases de datos relacionales y ejecución de consultas seguras
- Concepto y utilidad.
- Manejo de excepciones.
- Implementación de excepciones
- Jerarquía de excepciones.

Módulo 7. Interfaz gráfica y aplicación de pruebas

- Interfaces gráficas
- Pruebas de caja negra.
- Pruebas de unidad.
- Revisiones de escritorio.
- Revisión grupal de diseños orientados a objetos.

5. Metodología de enseñanza y aprendizaje

En el presente curso, se utilizan una serie de metodologías de enseñanza que se describen a continuación:

- La metodología de la enseñanza directa permite que el profesor utilice modelaciones sobre el diseño de aplicaciones de software orientados a objetos para que los estudiantes desarrollen programas computacionales que faciliten la solución de problemas.

- La metodología de aprendizaje autónomo fomenta en los estudiantes la responsabilidad y la autodirección en los procesos de aprendizaje para el desarrollo de los trabajos académicos relacionados con resolución de problemas por medio de programación.
- La metodología del aprendizaje basado en proyectos facilita en los alumnos el desarrollo de las etapas que se requieren para la creación de aplicaciones de software mediante un lenguaje de alto nivel y el paradigma orientado a objetos.
- La metodología de aprendizaje colaborativo promueve que los estudiantes discutan ideas entre pares para el análisis, el diseño y la construcción de aplicaciones de software que faciliten el desarrollo de soluciones a problemas de alta complejidad computacional.
- La metodología de aprendizaje interactivo posibilita en los estudiantes el intercambio de ideas con sus pares para la exploración de soluciones a problemas de contextos reales relacionados con la programación de computadoras.
- La guía del docente favorece que el profesor aclare dudas e interactúe con los estudiantes sobre las estructuras de la programación de aplicaciones para el desarrollo de mejores técnicas en la elaboración de código basado en el paradigma de orientación a objetos.
- La metodología de evaluación permite la implementación de evaluaciones sumativas mediante la creación de una aplicación de software, la resolución de prácticas de programación y un portafolio de evidencias; evaluaciones formativas por medio de estrategias de aprendizaje como la elaboración de mapas conceptuales y la esquematización de los requerimientos de software; y evaluaciones diagnósticas que permitan la identificación de los

conocimientos previos de los estudiantes sobre el paradigma de orientación a objetos.

6. Estrategias de aprendizaje

Las estrategias de aprendizaje que se emplean para este curso se describen a continuación:

Estrategias sumativas:

- El desarrollo de la aplicación de software permite a los estudiantes la integración de conocimientos de programación orientada a objetos y bases de datos para la creación de un programa complejo, mediante el diseño de arquitectura de software, la creación de interfaces de usuario y la resolución de problemas.
- La compleción de prácticas de programación facilita a los estudiantes la aplicación de técnicas de programación orientada a objetos y consulta de bases de datos para la resolución de problemas y el desarrollo de habilidades de codificación.
- La elaboración del portafolio de evidencias propicia en los estudiantes la organización de los conocimientos sobre programación orientada a objetos, arquitectura de software e interfaces de usuario, por medio de la investigación y la síntesis de las temáticas asignadas.

Estrategias formativas:

- La esquematización de los requerimientos de software les permite a los estudiantes la distinción de las herramientas que se necesitan para la creación de aplicaciones de software.
- La elaboración de mapas mentales sobre el lenguaje de modelado unificado propicia que los estudiantes identifiquen las características que deben

cumplir la arquitectura de las aplicaciones que se crean con programación de clases y objetos.

7. Sistema de evaluación

A continuación, se describen las actividades de aprendizaje correspondientes a la evaluación sumativa:

Actividades de aprendizaje	Porcentaje
Aplicación de software (1)	50%
Prácticas de programación (2)	30%
Portafolio de evidencias (1)	20%
Total	100%

Aplicación de software (50%). La aplicación de software consiste en el desarrollo de un programa de computadora complejo que integra la programación orientada a objetos y la consulta de bases de datos. Los estudiantes deben diseñar la arquitectura de software, los modelados, las interfaces de usuario y la consulta de bases de datos dentro del entorno de programación de alto nivel. Esta actividad de evaluación consta de dos entregables parciales y uno final, los cuales se desarrollan de manera grupal y se presentan según se describe en el cronograma del curso. Las secciones y aspectos que se deben tener en cuenta en el desarrollo del proyecto se describen en la siguiente rúbrica:

Criterios de desempeño	No logra el criterio (1 punto)	Logra parcialmente el criterio (2 puntos)	Logra el criterio (4 puntos)	Excede el criterio (5 puntos)

Repositorio de la aplicación para la creación inicial	Desarrolla el repositorio en una aplicación de versiones de manera inadecuada	Desarrolla el repositorio en una aplicación de versiones de manera básica	Desarrolla el repositorio en una aplicación de versiones de manera regular	Desarrolla el repositorio en una aplicación de versiones de manera adecuada
Diagrama de lenguaje unificado básico	Confecciona el diagrama de lenguaje de modelado básico de manera inadecuada e incluye las clases y parámetros erróneamente	Confecciona el diagrama de lenguaje de modelado básico de manera elemental e incluye las clases y parámetros con varios errores	Confecciona el diagrama de lenguaje de modelado básico de manera regular e incluye las clases y parámetros con algún error	Confecciona el diagrama de lenguaje de modelado básico de manera suficiente e incluye las clases y parámetros de manera apropiada
Estructura inicial del código	Elabora la estructura del código en un lenguaje de alto nivel de forma muy básica y presenta múltiples errores	Elabora la estructura del código en un lenguaje de alto nivel de forma básica y presenta varios errores	Elabora la estructura del código en un lenguaje de alto nivel de forma regular y presenta algún error	Elabora la estructura del código en un lenguaje de alto nivel de forma adecuada y no presenta errores
Diagrama de lenguaje unificado final	Confecciona el diagrama de lenguaje de modelado final de	Confecciona el diagrama de lenguaje de modelado final de manera elemental e incluye las	Confecciona el diagrama de lenguaje de modelado final de manera regular e incluye las	Confecciona el diagrama de lenguaje de modelado final de manera suficiente e

	manera inadecuada e incluye las clases y parámetros erróneamente	clases y parámetros con varios errores	clases y parámetros con algún error	incluye las clases y parámetros de manera suficiente
Repositorio de la aplicación para la lógica de clases	Desarrolla el repositorio de la aplicación e para la lógica de clases de manera inadecuada	Desarrolla el repositorio de la aplicación para la lógica de clases de manera básica	Desarrolla el repositorio de la aplicación para la lógica de clases de manera regular	Desarrolla el repositorio de la aplicación para la lógica de clases de manera adecuada
Interfaz gráfica de la aplicación de la pantalla de registro	Elabora la interfaz gráfica de la aplicación de la pantalla de registro de forma básica y con errores en el patrón de objetos de acceso	Elabora la interfaz gráfica de la aplicación de la pantalla de registro de forma elemental y con algunos errores en el patrón de objetos de acceso	Elabora la interfaz gráfica de la aplicación de la pantalla de registro de forma regular y con algún error en el patrón de objetos de acceso	Elabora la interfaz gráfica de la aplicación de la pantalla de registro de forma adecuada y sin errores en el patrón de objetos de acceso
Relación de la interfaz gráfica y la lógica de las clases en Java	Relaciona la interfaz gráfica y la lógica de las clases en Java de manera inadecuada	Relaciona la interfaz gráfica y la lógica de las clases en Java de manera básica	Relaciona la interfaz gráfica y la lógica de las clases en Java de manera regular	Relaciona la interfaz gráfica y la lógica de las clases en Java de manera exacta

Arquitectura lógica de las clases e interfaz de la aplicación	Entrega la arquitectura de la lógica de clases e interfaz de la aplicación de forma muy básica y con errores en el patrón de objetos de acceso de datos	Entrega la arquitectura de la lógica de clases e interfaz de la aplicación de forma básica y con algunos errores en el patrón de objetos de acceso de datos	Entrega la arquitectura de la lógica de clases e interfaz de la aplicación de forma regular y con algún error en el patrón de objetos de acceso de datos	Entrega la arquitectura de la lógica de clases e interfaz de la aplicación de forma suficiente y sin errores en el patrón de objetos de acceso de datos
Lectura de credenciales y el servido de base de datos	Implementa la arquitectura de la lógica e interfaz de usuario de la aplicación de forma escasa y con errores en el patrón de objetos de acceso de datos	Implementa la arquitectura de la lógica e interfaz de usuario de la aplicación de forma básica y con algunos errores en el patrón de objetos de acceso de datos	Implementa la arquitectura de la lógica e interfaz de usuario de la aplicación de forma regular y con algún error en el patrón de objetos de acceso de datos	Implementa la arquitectura de la lógica e interfaz de usuario de la aplicación de forma adecuada y sin errores en el patrón de objetos de acceso de datos
Funcionalidad de la aplicación con casos de prueba	Contiene uno o dos casos presentados de forma inadecuada.	Contiene solo un caso de prueba completo y correcto. El otro caso de prueba está ausente o incompleto.	Contiene dos casos de prueba, pero presentan algún error.	Contiene dos casos de prueba completos y adecuados.

Explicación del programa computacional	Realiza una explicación del programa de forma escueta y contesta de forma inadecuada las preguntas hechas por el docente	Realiza una explicación del programa de forma básica y contesta de forma básica varias de las preguntas hechas por el docente	Realiza una explicación del programa de forma regular y contesta alguna de las preguntas hechas por el docente de forma errónea	Realiza una explicación del programa de forma adecuada y contesta adecuadamente las preguntas hechas por el docente
Retroalimentación del docente	Incorpora la retroalimentación brindada por el docente de forma incoherente y escasa	Incorpora la retroalimentación brindada por el docente de manera básica	Incorpora la retroalimentación brindada por el docente de manera incompleta	Incorpora la retroalimentación brindada por el docente de manera clara y completa
Total de puntos: 60	**Puntos obtenidos:**		**Porcentaje obtenido:**	
Retroalimentación:				

Prácticas de programación (30%, 15% cada una). Las prácticas de programación facilitan a los estudiantes profundizar en las técnicas y procesos utilizados en la programación orientada a objetos y la consulta de bases de datos. Los estudiantes desarrollan y recopilan el código fuente de un programa informático que brinda la solución a ejercicios especificados por el docente. Esta actividad se realiza de manera individual, el entregable consiste en un documento escrito y un archivo de programación con el desarrollo de la solución de la práctica, los cuales se presentan según se describe en el cronograma del curso.

Las secciones y aspectos que debe contener la solución de la práctica se describen en la siguiente rúbrica:

Criterios de desempeño	No logra el criterio (1 punto)	Logra parcialmente el criterio (2 puntos)	Logra el criterio (4 puntos)	Excede el criterio (5 puntos)
Repositorio de la aplicación para la creación inicial	Desarrolla el repositorio en una aplicación de versiones de manera inadecuada	Desarrolla el repositorio en una aplicación de versiones de manera básica	Desarrolla el repositorio en una aplicación de versiones de manera regular	Desarrolla el repositorio en una aplicación de versiones de manera adecuada
Código en el lenguaje de alto nivel	Implementa los códigos de el desarrollo de la práctica de manera inadecuada y con múltiples errores	Implementa los códigos de el desarrollo de la práctica de manera básica y con algunos errores	Implementa los códigos de el desarrollo de la práctica de manera regular y con algún error	Implementa los códigos de el desarrollo de la práctica de manera adecuada y sin errores
Procesos de la orientación de objetos de la práctica	Elabora un documento que explica los procesos de abstracción, relación de atributos y objetos de la práctica de manera escueta	Elabora un documento que explica los procesos de abstracción, relación de atributos y objetos de la práctica de manera básica	Elabora un documento que explica los procesos de abstracción, relación de atributos y objetos de la práctica de manera regular	Elabora un documento que explica los procesos de abstracción, relación de atributos y objetos de la práctica de manera adecuada
Repositorio de la aplicación para la lógica de clases	Elabora el repositorio de la aplicación para la lógica de clases de	Elabora el repositorio de la aplicación para la lógica de clases de	Elabora el repositorio de la aplicación para la lógica de clases de	Elabora el repositorio de la aplicación para la lógica de clases de

	manera inadecuada	manera básica	manera regular	manera adecuada
Librería de la interfaz de usuario	Genera la librería de la capa lógica y la importación en la interfaz de usuario de forma inadecuada y con errores	Genera la librería de la capa lógica y la importación en la interfaz de usuario de forma básica y con algunos errores	Genera la librería de la capa lógica y la importación en la interfaz de usuario de forma regular y con algún error	Genera la librería de la capa lógica y la importación en la interfaz de usuario de forma adecuada y sin errores
Arquitectura de software: registro, enlistar y datos	Diseña una arquitectura de software de forma inapropiada, y no permite las funcionalidades del usuario	Diseña una arquitectura de software de forma básica, y permite menos de la mitad de las funcionalidades del usuario	Diseña una arquitectura de software de forma regular, y permite más de la mitad de las funcionalidades del usuario	Diseña una arquitectura de software de forma adecuada, y permite todas las funcionalidades del usuario
Interfaz gráfica de la aplicación	Elabora la interfaz gráfica de la aplicación de forma muy básica y con errores en el patrón de objetos de acceso	Elabora la interfaz gráfica de la aplicación forma básica y con algunos errores en el patrón de objetos de acceso	Elabora la interfaz gráfica de la aplicación de forma regular y con algún error en el patrón de objetos de acceso	Elabora la interfaz gráfica de la aplicación de forma adecuada y sin errores en el patrón de objetos de acceso
Entrega de la solución de la práctica	Realiza la entrega de las secciones de la práctica de forma insuficiente			Realiza la entrega de la práctica con todas las secciones de forma adecuada

Total de puntos: 40	Puntos obtenidos:	Porcentaje obtenido:
Retroalimentación:		

Portafolio de evidencias (20%). El portafolio de evidencias consiste en la elaboración de un documento que incluya los conceptos, las definiciones y los métodos que se ven durante el curso sobre la programación orientada objetos, la arquitectura de software y las interfaces de aplicaciones. Los estudiantes por medio de una aplicación de recolección de información elaboran una descripción, interpretación y ejemplos de cada una de estas estructuras. Esta actividad se desarrolla de forma individual y consta de un entregable que se presentan según se describe en el cronograma del curso y debe abarcar los aspectos que se señalan en la siguiente rúbrica:

Criterios de desempeño	No logra el criterio (1 punto)	Logra parcialmente el criterio (2 puntos)	Logra el criterio (4 puntos)	Excede el criterio (5 puntos)
Recopilación de la información	Elabora las secciones del portafolio de forma desordena	Elabora las secciones del portafolio (en su minoría) de forma ordenada	Elabora las secciones del portafolio (en su mayoría) de forma ordenada	Elabora las secciones del portafolio de forma ordena
Elementos de multimedia	Incluye elementos de multimedia en el portafolio de manera limitada y confusa	Incluye elementos de multimedia en el portafolio de manera básica y poco clara	Incluye elementos de multimedia en el portafolio de manera regular y clara	Incluye elementos de multimedia en el portafolio de manera adecuada y explícita

Técnicas y métodos de la programación orientada a objetos	Incorpora en el portafolio las técnicas y métodos de la programación orientada a objetos de forma escueta	Incorpora en el portafolio las técnicas y de la programación orientada a objetos de forma básica	Incorpora en el portafolio las técnicas y métodos de la programación orientada a objetos de forma regular	Incorpora en el portafolio las técnicas y métodos de la programación orientada a objetos de forma suficiente
Estructuras de datos y arquitectura de software	Incorpora en el portafolio las estructuras de datos y arquitectura de software forma escueta	Incorpora en el portafolio las estructuras de datos y arquitectura de software de forma básica	Incorpora en el portafolio las estructuras de datos y arquitectura de software de forma regular	Incorpora en el portafolio las estructuras de datos y arquitectura de software de forma adecuada
Ejemplos de las estructuras	Integra al portafolio ejemplos de las estructuras de programación y arquitectura de manera inadecuada	Integra al portafolio ejemplos de las estructuras de programación y arquitectura de manera básica	Integra al portafolio ejemplos de las estructuras de programación y arquitectura de manera aceptable	Integra al portafolio ejemplos de las estructuras de programación y arquitectura de manera adecuada
Entrega del portafolio	Entrega el portafolio de forma inaceptable			Entrega el portafolio de forma adecuada
Total de puntos: 30	**Puntos obtenidos:**		**Porcentaje obtenido:**	
Retroalimentación:				

8. Recursos didácticos

Para promover el aprendizaje de los estudiantes, se incorporan los siguientes recursos educativos:

- El conocimiento y la experiencia previa sobre el paradigma de orientación a objetos permiten que los estudiantes reconozcan las estructuras y las arquitecturas de las aplicaciones y programas de software.
- El repositorio de acceso libre Bitbucket permite el alojamiento de los códigos realizados por los estudiantes en las diferentes actividades del curso, también promueve el trabajo colaborativo en la creación de aplicaciones de software.
- La aplicación IntelliJ IDEA es un entorno de desarrollo integrado (IDE)se utiliza como recurso de apoyo para la implementación de la aplicación de computadora que se realiza en el curso en el lenguaje de programación Java.
- El lenguaje de programación Java se utiliza como recurso de apoyo para el desarrollo de los programas de computadora realizados en el curso por medio de las estructuras de secuencia, condicional e iterativas como también las orientadas a objetos.
- Los estándares ISO/IEC/IEEE 24765 e ISO/IEC/IEEE 12207 son marcos internacionales que establecen el vocabulario de sistemas e ingeniería de software y los procesos del ciclo de vida del software, respectivamente, lo cual proporciona a los estudiantes un lenguaje técnico estandarizado y una estructura metodológica para comprender las fases clave del desarrollo de software.
- La Guía sobre el Cuerpo de Conocimientos de Ingeniería del Software (SWEBOK v.4) (IEEE) define las áreas clave del conocimiento en ingeniería del software reconocidas por la comunidad profesional, lo que permite

estructurar los contenidos del curso conforme a prácticas estandarizadas, incluir resultados de aprendizaje relevantes y orientar la formación hacia competencias demandadas en la industria del software.

- Las Orientaciones Curriculares para Planes de Estudios en Ingeniería del Software (SE2014) (ACM, IEEE-CS) establece los conocimientos fundamentales, las habilidades y las actitudes que deben adquirir los estudiantes en un programa universitario de ingeniería del software, esto permite incluir en el curso áreas temáticas, resultados de aprendizaje y recomendaciones pedagógicas alineadas con las necesidades de la industria del software.
- La biblioteca de la Universidad, Ignacio Trejos Zelaya, es el recurso bibliográfico con el que cuentan los docentes y estudiantes para realizar consultas de libros, revistas, tesis y manuales tanto físicos como digitales.
- El servicio con licenciamiento de Google Workspace for Education permite a los estudiantes y docentes contar con diversos productos como espacios de almacenamiento en la nube, herramientas de ofimática y aplicaciones de comunicación sincrónica y asincrónica que apoyan los procesos de enseñanza y aprendizaje.

9. Bibliografía

Bibliografía obligatoria

Domínguez, T. (2024). *JavaScript como nunca antes se lo habían contado* (1.ª ed.). Marcombo. https://elibro.net/es/lc/ucenfotec/titulos/281752

Fernández, P. (2023). *Creación de componentes en JavaScript: Curso práctico* (1.ª ed.). RA-MA Editorial. https://elibro.net/es/lc/ucenfotec/titulos/235061

Jiménez, C. (2021). *UML: Arquitectura de aplicaciones en Java, C++ y Python* (2.ª ed.). RA-MA Editorial. https://elibro.net/es/ereader/ucenfotec/222720

Bibliografía complementaria

Aristizábal Martínez, D. A., y Quiceno Metaute, S. M. (2023). *Lógica de programación básica orientada a objetos con ejercicios resueltos* (1.ª ed.). Instituto Tecnológico Metropolitano. https://elibro.net/es/lc/ucenfotec/titulos/253798

Martín, A., y Rubio, M. (2021). *Lenguajes de programación*. UNED - Universidad Nacional de Educación a Distancia. https://elibro.net/es/ereader/ucenfotec/184827

Vegas, J. (2021). *Java 17: Programación avanzada* (1.ª ed.). RA-MA Editorial. https://elibro.net/es/ereader/ucenfotec/222668

10. Cronograma

Sesión	Módulo	Actividades y estrategias de aprendizaje
1	**Módulo 1. Conceptos del paradigma de programación orientado a objetos** • Abstracción. • Encapsulamiento. • Modularidad.	- Realiza la lectura de la unidad 8 de Domínguez, T. (2024): https://elibro.net/es/lc/ucenfotec/titulos/281752 - Participa en la sesión de clases de la semana con el docente - Asiste a las diferentes actividades de inducción estudiantil de la universidad

		- Realiza la revisión de software y herramientas tecnológicas que se utilizan durante el curso
2		- Realiza la lectura de la unidad 8 de Domínguez, T. (2024): https://elibro.net/es/lc/ucenfotec/titulos/281752 - Participa en la sesión de clases de la semana con el docente - **Aplicación de software:** Empieza el desarrollo del **primer avance** de la aplicación de software y consulta posibles dudas al docente
3	**Módulo 2. Relaciones entre objetos** ● Dependencia. ● Asociación. ● Agregación. ● Composición. ● Representación de relaciones	- Realiza la lectura del capítulo 4 de Fernández, P. (2023): https://elibro.net/es/lc/ucenfotec/titulos/235061 - Participa en la sesión de clases de la semana con el docente - **Aplicación de software:** Continúa el desarrollo del **primer avance** de la aplicación de software y consulta posibles dudas al docente
4		- Realiza la lectura del capítulo 4 de Fernández, P. (2023): https://elibro.net/es/lc/ucenfotec/titulos/235061 - Participa en la sesión de clases de la semana con el docente - **Aplicación de software (15%):** Finaliza el desarrollo del **primer**

		avance de la aplicación de software y lo entrega según las indicaciones del docente
5	**Módulo 3. Relaciones entre clases** ● Estructura de clases. ● Herencia y redefinición de métodos. ● Modificadores de acceso. ● Clases abstractas y polimorfismo. ● Ligado: estático y dinámico. ● Conversión de tipos	- Realiza la lectura del capítulo 15, sección 2 de Jiménez, C. (2021): https://elibro.net/es/ereader/ucenfotec/222720 - Participa en la sesión de clases de la semana con el docente - **Práctica de programación (15%):** Realiza el desarrollo de la **primera** práctica de programación y lo entrega según las indicaciones del docente
6		- Realiza la lectura del capítulo 6 de Fernández, P. (2023): https://elibro.net/es/lc/ucenfotec/titulos/235061 - Participa en la sesión de clases de la semana con el docente - **Portafolio de evidencias:** Inicia el desarrollo del **primer entregable** del portafolio y consulta posibles dudas al docente
7	**Módulo 4. Modelado de aplicaciones de software** ● Técnicas de Modelado. ● Contrato de Operaciones. ● Diagramas de interacción. ● Diagramas de Actividad.	- Realiza la lectura del capítulo 6 de Jiménez, C. (2021): https://elibro.net/es/ereader/ucenfotec/222720 - Participa en la sesión de clases de la semana con el docente - **Portafolio de evidencias (10%):** Finaliza el desarrollo del **primer entregable** del portafolio y lo entrega según las indicaciones del docente

8		- Realiza la lectura del capítulo 7 de Jiménez, C. (2021): https://elibro.net/es/ereader/ucenfotec/222720 - Participa en la sesión de clases de la semana con el docente - **Práctica de programación (15%):** Realiza el desarrollo de la **segunda** práctica de programación y lo entrega según las indicaciones del docente
9	**Módulo 5. Interfaces, expresiones e hilos** ● Concepto, terminología e implementación. ● Comparación con clases abstractas. ● Comparación con herencia. ● Manejo de expresiones e hilos	- Realiza la lectura del capítulo 8 de Jiménez, C. (2021): https://elibro.net/es/ereader/ucenfotec/222720 - Participa en la sesión de clases de la semana con el docente - **Aplicación de software:** Empieza el desarrollo del **segundo avance** de la aplicación de software y consulta posibles dudas al docente
10		- Realiza la lectura del capítulo 9 de Jiménez, C. (2021): https://elibro.net/es/ereader/ucenfotec/222720 - Participa en la sesión de clases de la semana con el docente - **Aplicación de software (15%):** Finaliza el desarrollo del **segundo avance** de la aplicación de software y lo entrega según las indicaciones del docente

11	**Módulo 6. Arquitectura de software y excepciones** ● Arquitectura de múltiples capas ● Conexión a bases de datos relacionales y ejecución de consultas seguras ● Concepto y utilidad. ● Manejo de excepciones. ● Implementación de excepciones ● Jerarquía de excepciones.	- Realiza la lectura del capítulo 11 de Jiménez, C. (2021): https://elibro.net/es/ereader/ucenfotec/222720 - Participa en la sesión de clases de la semana con el docente - **Portafolio de evidencias:** Empieza el desarrollo del **segundo entregable** del portafolio y consulta posibles dudas al docente
12		- Realiza la lectura del capítulo 12 de Jiménez, C. (2021): https://elibro.net/es/ereader/ucenfotec/222720 - Participa en la sesión de clases de la semana con el docente - **Portafolio de evidencias (10%):** Finaliza el desarrollo del **segundo entregable** del portafolio y lo entrega según las indicaciones del docente
13	**Módulo 7. Interfaz gráfica y aplicación de pruebas** ● Interfaces gráficas ● Pruebas de caja negra. ● Pruebas de unidad. ● Revisiones de escritorio. ● Revisión grupal de diseños orientados a objetos.	- Realiza la lectura de la unidad 13 de Domínguez, T. (2024): https://elibro.net/es/lc/ucenfotec/titulos/281752 - Participa en la sesión de clases de la semana con el docente - **Aplicación de software:** Empieza el desarrollo de la **entrega final** de la aplicación de software y consulta posibles dudas al docente

14		- Realiza la lectura del capítulo 3 de Fernández, P. (2023): https://elibro.net/es/lc/ucenfotec/titulos/235061 - Participa en la sesión de clases de la semana con el docente - **Aplicación de software (20%):** Finaliza el desarrollo de la **entrega final** de la aplicación de software y lo entrega según las indicaciones del docente
15	**Actividades finales del curso**	Se entregan los promedios finales del curso