




PROGRAMA DE ESTUDIOS

PROGRAMACIÓN ORIENTADA A OBJETOS I
TECNÓLOGO EN DESARROLLO DE SOFTWARE

TERCER SEMESTRE
EDUCACIÓN MEDIA SUPERIOR





Programación Orientada a Objetos I. Programa de Estudios. Tecnólogo en Desarrollo de Software. Tercer Semestre, fue editado por el Centro de Enseñanza Técnica Industrial de Jalisco.

LETICIA RAMÍREZ AMAYA
Secretaria de Educación Pública

CARLOS RAMÍREZ SÁMANO
Subsecretario de Educación Media Superior

JUDITH CUÉLLAR ESPARZA
Directora General del Centro de Enseñanza Técnica Industrial


EMMA DEL CARMEN ALVARADO ORTIZ
Directora Académica del Centro de Enseñanza Técnica Industrial

Primera edición, 2024.

D. R. © CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL. ORGANISMO PÚBLICO
DESCENTRALIZADO FEDERAL.

Nueva Escocia No. 1885, Col. Providencia 5ª sección, C. P. 44638, Guadalajara,
Jalisco.

Distribución gratuita. Prohibida su venta.



ÍNDICE

06

I. IDENTIFICACIÓN DEL CURSO

07

II. UBICACIÓN DE LA UAC

09

III. DESCRIPTORES DE LA UAC

11

IV. DESARROLLO DE LA UAC

15

V. RECURSOS BIBLIOGRÁFICOS Y
OTRAS FUENTES DE CONSULTA

PRESENTACIÓN



El rediseño curricular del modelo educativo del tecnólogo, articula los tres componentes del Marco Curricular Común de la Educación Media Superior: i) el fundamental; ii) el ampliado; y iii) el profesional, ahora laboral, conservando este último, el enfoque basado en competencias, bajo una nueva propuesta que impulsa al CETI a mantener una estrecha vinculación con el sector productivo. El planteamiento del proceso educativo surge a partir del campo profesional, lo que permite diseñar la situación didáctica desde una problemática que pone en juego e integra las competencias del estudiantado para la transformación laboral y el aprendizaje significativo dejando a un lado, la idea del empleo.

En este sentido, la presente asignatura plantea desde su propia construcción, un proyecto integrador que va orientando el perfil de egreso y que hace explícito los conocimientos, destrezas, habilidades, actitudes y valores que las y los estudiantes aplican en los procedimientos técnicos específicos.

La unidad de aprendizaje de Programación Orientada a Objetos I es un componente esencial en la formación de estudiantes y futuros tecnólogos profesionales en las áreas de Desarrollo de Software y Sistemas de Información. Su propósito principal radica en preparar a las y los estudiantes para que desarrollen competencias en el desarrollo de software aplicando los principios y elementos del paradigma de programación orientada a objetos que les permita resolver problemas y desarrollar soluciones informáticas eficientes. La intencionalidad detrás de esta unidad de aprendizaje es fortalecer las habilidades en la abstracción, modelado, diseño y programación orientada a objetos que les permita comprender y aplicar nuevas herramientas y modelos de programación para desarrollar aplicaciones de mayor alcance como parte de equipos de trabajo de desarrollo de software así como para buscar que se desarrolle la habilidad de abordar y resolver problemas con un enfoque basado en requerimientos específicos en su ámbito académico y personal. La meta es que las y los estudiantes puedan tomar los requerimientos de un cliente o las necesidades específicas de una situación, diseñar sus diagramas de clases y algoritmos e implementarlos en un lenguaje de alto nivel con un enfoque orientado a objetos y desarrollando interfaces a la medida de las necesidades y requerimientos, así como el uso de buenas prácticas de programación, lo que implica no solo crear soluciones funcionales, sino también orientadas al usuario final, robustas y eficientes; y por última instancia, la unidad de aprendizaje tiene como objetivo formar profesionales en Desarrollo de Software con capacidad crítica, ética, innovación y responsabilidad social.

I. IDENTIFICACIÓN DEL CURSO

CARRERA: TECNÓLOGO EN DESARROLLO DE SOFTWARE

Modalidad:
Presencial

UAC:
Programación Orientada
a Objetos I

Clave:
233bMCLDS0301

Semestre:
Tercero

Academia:
Computación

Línea de Formación:
Desarrollo de aplicaciones

Créditos:
7.2

Horas Semestre:
72

Horas Semanales:
4

Horas Teoría:
2

Horas Práctica:
2

Fecha de elaboración:
Mayo 2024

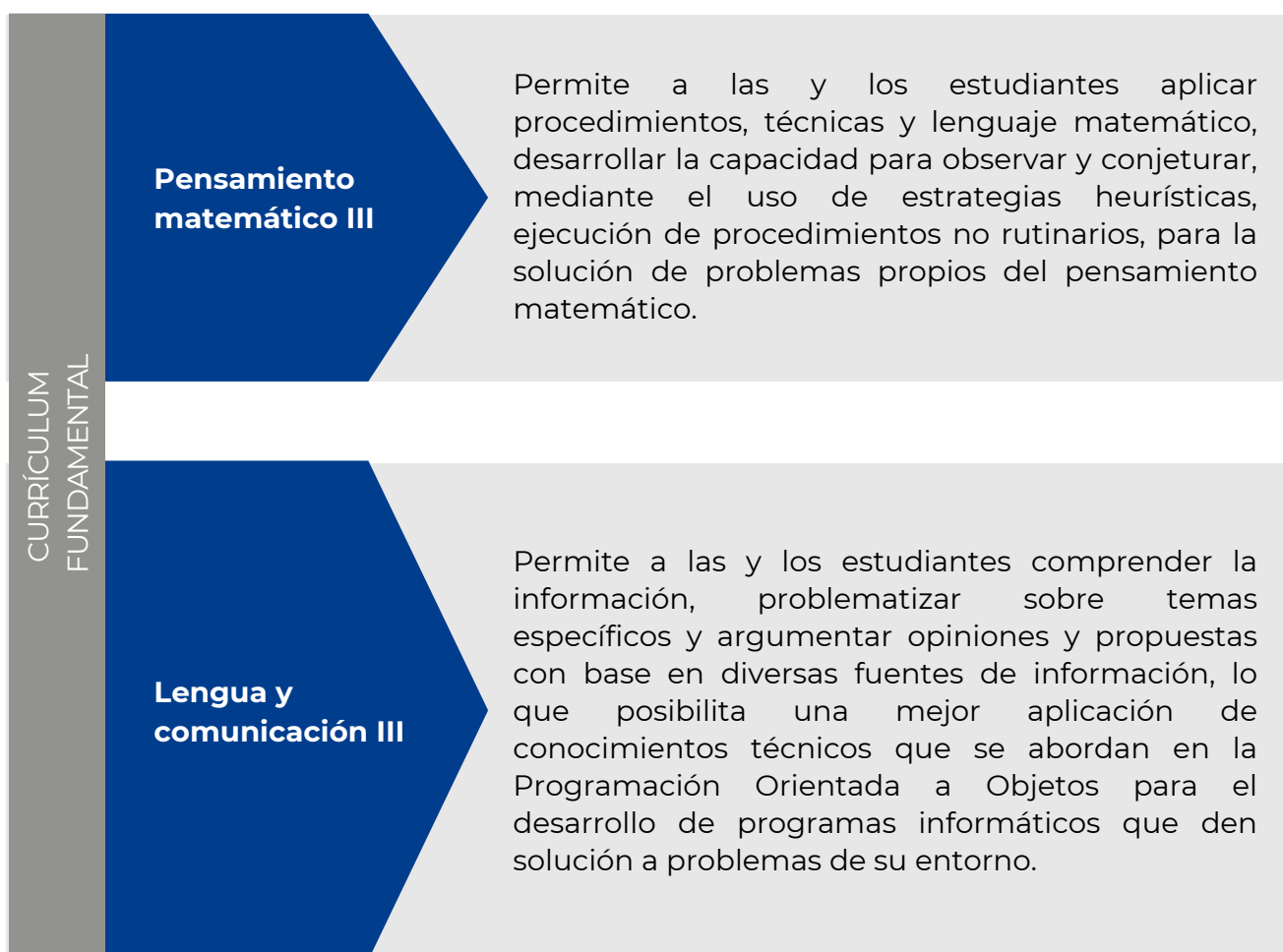
Fecha de última actualización:

II. UBICACIÓN DE LA UAC

ÁMBITOS DE TRANSVERSALIDAD

Relación con asignaturas respecto a Marco Curricular Común de Educación Media Superior (MCCEMS), es decir, currículum fundamental y con asignaturas del currículum laboral.

Asignaturas vinculadas / Tercer semestre



Asignatura previa / Segundo semestre

Se aplicaron los conceptos de modularidad, tipos de datos abstractos y funciones, los cuales son la base para la creación de los objetos, pues análogamente los datos abstractos representan la unificación de la información tal como son las propiedades de un objeto, la modularidad permite la creación de funciones y procedimientos los cuales forman los métodos de un objeto, componentes claves en la creación de clases. Estos conceptos son principios básicos para la abstracción y encapsulación de la programación orientada a objetos.

**Programación
estructurada**

CURRÍCULUM LABORAL

Asignatura posterior / Cuarto semestre

**Programación
orientada a objetos II**

Se adquirirán las bases sobre el uso de los mecanismos y características del paradigma orientado a objetos, el cual es el fundamento para la programación que emplean diversas estructuras de datos, algoritmos de ordenación y búsqueda, la representación abstracta de datos, la implementación de objetos genéricos, la programación concurrente y el uso de mecanismos de comunicación y operación en red que se abordan en Programación Orientada a Objetos II.

La UAC de Programación Orientada a Objetos I, permite desarrollar e interiorizar en las y los estudiantes los conceptos y características del Paradigma de la programación Orientada a Objetos tales como clases, métodos, atributos, objetos, constructores, herencia y polimorfismo, diseño modular e interfaces gráficas para el desarrollo de aplicaciones informáticas con enfoque al usuario final.

III. DESCRIPTORES DE LA UAC

1. META DE APRENDIZAJE DE LA ASIGNATURA

Aplica los mecanismos y características del paradigma de la Programación Orientada a Objetos para el desarrollo de programas informáticos a partir de requerimientos específicos de situaciones y problemas del entorno mediante el uso de interfaces gráficas.

2. COMPETENCIAS LABORALES DE LA UAC

- Diseña clases y objetos basados en UML para desarrollar aplicaciones informáticas, atendiendo a los requerimientos específicos de una micro o pequeña empresa, con compromiso y responsabilidad hacia la calidad del producto final.
- Desarrolla interfaces gráficas para implementar programas informáticos eficientes, resolviendo situaciones o problemas basados en requerimientos específicos con enfoque al usuario final, demostrando empatía y orientación al usuario.
- Desarrolla programas informáticos utilizando los mecanismos y características del paradigma orientado a objetos, asegurando que contengan elementos que proporcionen al usuario final una experiencia y uso intuitivo, mostrando creatividad e innovación en las soluciones propuestas.
- Utiliza buenas prácticas de programación en el diseño de soluciones informáticas, implementando clases y objetos relacionados con las necesidades de una pequeña o microempresa, con ética profesional y respeto por los estándares de calidad del software.

3. PRODUCTO INTEGRADOR

Programa informático.

Reporte técnico del programa informático.

3.1 Descripción del Producto Integrador

Desarrollar un programa informático tipo aplicación de escritorio con interfaz gráfica a base de ventanas, botones, listas, cuadros de diálogo, etcétera; para la gestión de la información de las operaciones de una micro o pequeña empresa.

Reporte sobre el programa informático que contenga:

- Datos del estudiante.
- Descripción del problema a resolver.
- Análisis del problema a resolver.
- Modelado del sistema.

3.2 Formato de Entrega

- Código fuente y código ejecutable de un programa informático.
- Reporte en documento Word o PDF.



IV. DESARROLLO DE LA UAC

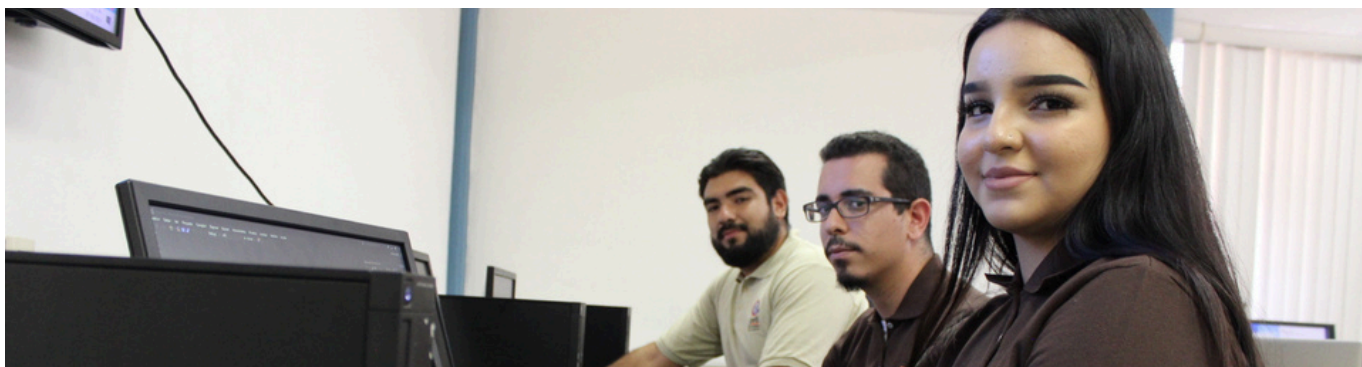
UNIDAD 1. INTRODUCCIÓN AL PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

Procesos	Contenidos	Recursos	Productos	Evaluación e instrumentos de evaluación
<p>Identifica los principios fundamentales de la POO.</p>	<ul style="list-style-type: none"> ● Análisis y abstracción de objetos. ● Definición de clases y objetos. ● Definición de los principios de la POO. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP1.1.1 Resumen con la definición de diversos conceptos y principios dentro de la POO. 	<ul style="list-style-type: none"> ● Lista de cotejo de los elementos que definen los principios y características.
<p>Aplica el análisis orientado a objetos para el diseño de clases y su representación gráfica mediante UML.</p>	<ul style="list-style-type: none"> ● Abstracción y representación gráfica de objetos en UML. ● Encapsulamiento y visibilidad. ● Métodos y atributos. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP1.2.1 Diagramas de clases base. 	<ul style="list-style-type: none"> ● Rúbrica de aplicación de reglas y simbología en la definición y representación de clases.

Procesos	Contenidos	Recursos	Productos	Evaluación e instrumentos de evaluación
<p>Identifica los distintos niveles de encapsulación para los elementos de una clase.</p>	<ul style="list-style-type: none"> ● Control de acceso a atributos y métodos. ● Uso del EID y la estructura de un programa. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP1.3.1 Diagramas de clases aplicando control de acceso. 	<ul style="list-style-type: none"> ● Rúbrica sobre la correlación entre un diagrama de clases y su programa equivalente.
<p>Aplica los principios básicos de la POO en ejercicios prácticos de diseño de clases y la creación y uso de objetos en un lenguaje de programación.</p>	<ul style="list-style-type: none"> ● Instancia de clases de biblioteca. ● Instancia de clases propias. ● Constructor por defecto. ● Uso y creación de métodos. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP1.4.1 Programas codificados a partir de diagramas de clases. 	<ul style="list-style-type: none"> ● Rúbrica de la implementación adecuada de clases a partir de su diagrama.
<p>Desarrolla un programa básico a partir de su diagrama de clases que implemente métodos constructores y métodos de propósito específico.</p>	<ul style="list-style-type: none"> ● Estructura de un proyecto de un programa orientado a objetos. ● Uso de diversos métodos. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP1.5.1 Reporte documentado del desarrollo de un proyecto para un programa orientado a objetos. 	<ul style="list-style-type: none"> ● Rúbrica sobre el desarrollo de un proyecto orientado a objetos.
<p>PPI: Reporte documentado que evidencia la creación de diagramas de clases mediante el lenguaje UML para visualizar, especificar, construir y documentar un programa informático.</p>				

UNIDAD 2. IMPLEMENTACIÓN DE LOS PRINCIPIOS FUNDAMENTALES DEL PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS

Procesos	Contenidos	Recursos	Productos	Evaluación e instrumentos de evaluación
<p>Implementa distintos niveles de encapsulación en una clase.</p>	<ul style="list-style-type: none"> • Visibilidad de los miembros de una clase. • Acceso a los miembros de una clase. 	<ul style="list-style-type: none"> • Equipo de cómputo. • Material audiovisual. • Entorno Integrado de Desarrollo. • Plataforma virtual de apoyo al aprendizaje. • Guía de actividades. 	<ul style="list-style-type: none"> • SP2.1.1 Reporte documentado con diagramas de clases. 	<ul style="list-style-type: none"> • Lista de cotejo del reporte documentado.
<p>Crea objetos a partir de clases definidas para una operación y propósito específico con diversos tipos de métodos.</p>	<ul style="list-style-type: none"> • Constructores. • Tipos de métodos. • Sobrecarga de métodos. 	<ul style="list-style-type: none"> • Equipo de cómputo. • Material audiovisual. • Entorno Integrado de Desarrollo. • Plataforma virtual de apoyo al aprendizaje. • Guía de actividades. 	<ul style="list-style-type: none"> • SP2.2.1 Reporte documentado que evidencie la aplicación de constructores, diversos tipos de métodos y su sobrecarga. 	<ul style="list-style-type: none"> • Lista de cotejo sobre el desarrollo y contenido del programa.



Procesos	Contenidos	Recursos	Productos	Evaluación e instrumentos de evaluación
<p>Identifica las relaciones entre un conjunto de clases para crear un diagrama de clases con sus relaciones.</p>	<ul style="list-style-type: none"> ● Dependencia. ● Asociación. ● Generalización. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP2.3.1 Programa que el uso de clases relacionadas. 	<ul style="list-style-type: none"> ● Lista de cotejo sobre el desarrollo y contenido del programa.
<p>Comprende la jerarquía de clases para la implementación de la herencia.</p>	<ul style="list-style-type: none"> ● Herencia. ● Clases abstractas. ● Subclases. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP2.4.1 Proyecto de aplicación en el que se evidencia la implementación de clases abstractas y herencia. 	<ul style="list-style-type: none"> ● Lista de cotejo del desarrollo del programa.

PP2: Reporte documentado que evidencie la implementación de clases y objetos aplicando las características del paradigma orientado a objetos.

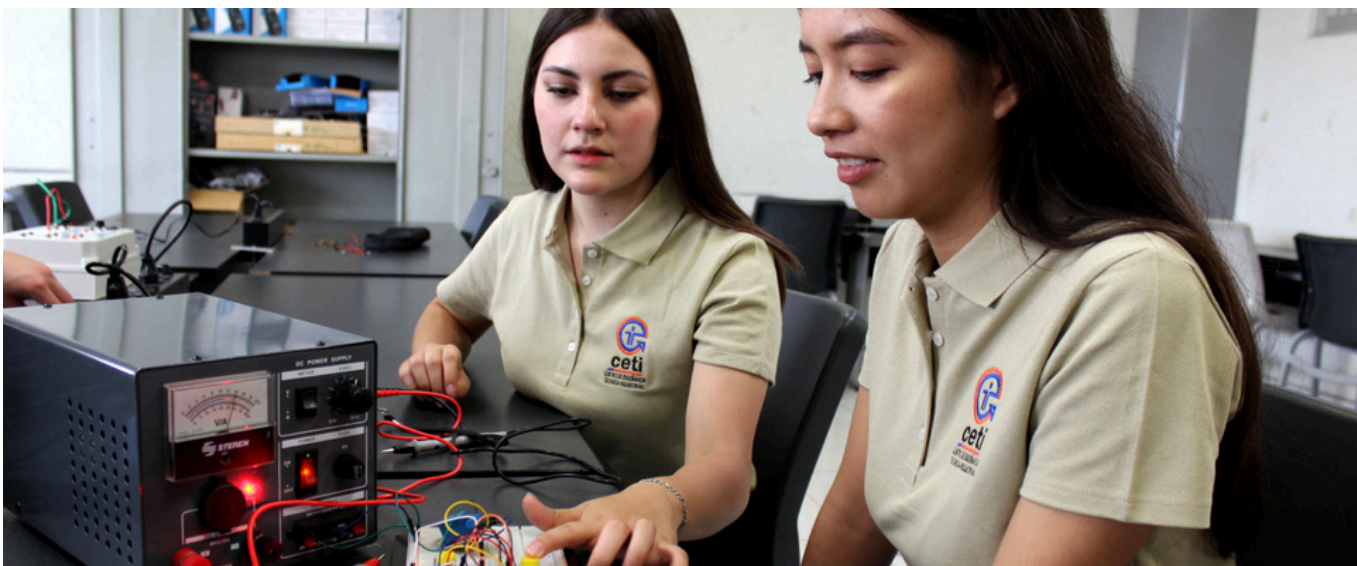


UNIDAD 3. DESARROLLO DE PROGRAMAS INFORMÁTICOS UTILIZANDO INTERFACES GRÁFICAS

Procesos	Contenidos	Recursos	Productos	Evaluación e instrumentos de evaluación
<p>Comprende los elementos para desarrollar un proyecto con interfaz gráfica.</p>	<ul style="list-style-type: none"> • Bibliotecas de clases de para GUI. • Formularios. • Elementos para programación de formularios. 	<ul style="list-style-type: none"> • Equipo de cómputo. • Material audiovisual. • Entorno Integrado de Desarrollo. • Plataforma virtual de apoyo al aprendizaje. • Guía de actividades. 	<ul style="list-style-type: none"> • SP3.1.1 Esquema gráfico (mockup). 	<ul style="list-style-type: none"> • Lista de cotejo para evaluación de mockups vs requerimientos.
<p>Utiliza elementos y controles para programar interfaces gráficas.</p>	<ul style="list-style-type: none"> • Controles intrínsecos de formularios. • Eventos. • Validaciones y excepciones. 	<ul style="list-style-type: none"> • Equipo de cómputo. • Material audiovisual. • Entorno Integrado de Desarrollo. • Plataforma virtual de apoyo al aprendizaje. • Guía de actividades. 	<ul style="list-style-type: none"> • SP3.2.1 Reporte documental sobre el uso de eventos, validaciones y excepciones. 	<ul style="list-style-type: none"> • Rúbrica sobre el uso de eventos y validación.

Procesos	Contenidos	Recursos	Productos	Evaluación e instrumentos de evaluación
<p>Utiliza diversos elementos gráficos para desarrollar programas orientados al usuario final.</p>	<ul style="list-style-type: none"> ● Contenedores y paneles. ● Componentes para el control de procesos y flujos de datos. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP3.3.1 Reporte documentado sobre el uso de contenedores de datos y controles. 	<ul style="list-style-type: none"> ● Rúbrica sobre el uso de contenedores de datos y controles.
<p>Implementa programas de propósito específico mediante interfaces gráficas.</p>	<ul style="list-style-type: none"> ● Bibliotecas de clases para GUI. ● Operaciones con flujos de datos. ● Referencias internas y externas. ● Publicación y empaquetado de aplicaciones. 	<ul style="list-style-type: none"> ● Equipo de cómputo. ● Material audiovisual. ● Entorno Integrado de Desarrollo. ● Plataforma virtual de apoyo al aprendizaje. ● Guía de actividades. 	<ul style="list-style-type: none"> ● SP3.4.1 Reporte documentado sobre bibliotecas, empaquetados, publicaciones y referencias. 	<ul style="list-style-type: none"> ● Rúbrica sobre los elementos del proceso del empaquetado de aplicaciones.

PF: Programa informático con interfaz para la gestión de información de las operaciones de una micro o pequeña empresa.



V. RECURSOS BIBLIOGRÁFICOS Y OTRAS FUENTES DE CONSULTA DE LA UAC

Recursos Básicos

- Alfonseca, M. Alcalá, A. (1992). *Programación Orientada a Objetos*. Madrid: Anaya Multimedia.
- Beck, K.; Cunningham, W. A. (1989). *Laboratory for teaching object-oriented thinking*. Proc. of Object-Oriented Programming Systems, Languages and Applications (OOPSLA '89). SIGPLAN Notices, Vol. 24, No. 10, October 89, pp 1-6.
- Meyer, B. (1999). *Construcción de software orientado a objetos*. México: Prentice Hall.
- Rubin, K.S.; Goldberg, A. (1992). *Object Behaviour Analysis*. *Comm. of the ACM*. vol. 35 no. 9, pp. 48-62, September 1992.

Recursos Complementarios

- Bloch, J. (2018). *Effective Java* (3rd ed.). Addison-Wesley.
- Deitel, P., & Deitel, H. (2017). *Java: How to Program* (10th ed.). Pearson.
- Skeet, J. (2019). *C# in Depth* (4th ed.). Manning Publications.
- Troelsen, A., & Japikse, P. (2017). *Pro C# 7: With .NET and .NET Core*. Apress.
- Wagner, B. (2018). *Effective C#: 50 Specific Ways to Improve Your C#* (3rd ed.). Addison-Wesley.

Fuentes de Consulta Utilizadas

- Cámara de Diputados del H. Congreso de la Unión. (30 de septiembre de 2019). Ley General de Educación. <https://www.diputados.gob.mx/LeyesBiblio/pdf/LGE.pdf>
- Diario Oficial de la Federación. (20 de septiembre de 2023). Acuerdo secretarial 17/08/22 y 09/08/23. https://www.dof.gob.mx/nota_detalle.php?codigo=5699835&fecha=25/08/2023
- Gobierno de México. (7 de septiembre de 2023). Propuesta del Marco Curricular Común de la Educación Media Superior. <https://educacionmediasuperior.sep.gob.mx/propuestaMCCEMS>

AGRADECIMIENTOS

El Centro de Enseñanza Técnica Industrial agradece al cuerpo docente por su participación en el diseño curricular:

Sergio Becerra Delgado

Sonia Erika Ibáñez de la Torre

Ismael López Buenrostro

María Luisa Díaz Rodríguez

Claudia Ureña Zapata

Patricia Elena Torres Álvarez

Alejandro Moreno Hernández

José Luis Roa García de la Paz

Angelberto Rosales Mayorga

Andrés Figueroa Flores

Juan Ramón Bravo López

Ana Elizabeth González Vásquez

Equipo Técnico Pedagógico

Armando Arana Valdez

Cynthia Isabel Zatarain Bastidas

Ciara Hurtado Arellano

Enrique García Tovar

Rodolfo Alberto Sánchez Ramos



Programación orientada a objetos I
Programa de estudios
Tecnólogo en Desarrollo de Software
Tercer Semestre



**GOBIERNO DE
MÉXICO**

