



GOBIERNO DE  
**MÉXICO**

EDUCACIÓN  
SECRETARÍA DE EDUCACIÓN PÚBLICA



Programa de estudios del módulo

# Programación básica orientada a objetos

## Currículum Laboral

Área:

Tecnología y transporte

Carrera:

Profesional Técnico-Bachiller en  
Soporte y mantenimiento de equipo de cómputo

5º semestre

**Editor:** Colegio Nacional de Educación Profesional Técnica

**Módulo:** Programación básica orientada a objetos

**Área:** Tecnología y transporte

**Carrera:** PT-B en Soporte y mantenimiento de equipo de cómputo.

**Semestre:** Quinto

**Horas por semana:** 5

**Fecha de diseño o actualización:** 28 de abril de 2025

**Vigencia:** a partir de la aprobación de la Junta Directiva y en tanto no se genere un documento que lo actualice.

© Colegio Nacional de Educación Profesional Técnica

Prohibida la reproducción total o parcial de esta obra por cualquier medio, sin autorización por escrito del CONALEP.

**Directorio**

**Rodrigo Alejandro Rojas Navarrete**  
Dirección General

**Ana María Rosas Muciño**  
Secretaría Académica

**Patricia Alejandra Bernal Monzón**  
Dirección de Diseño Curricular

## Programación básica orientada a objetos

Contenido	Pág.
<b>Capítulo I: Generalidades del Profesional Técnico-Bachiller</b>	
1.1 Marco Curricular Común de la Educación Media Superior	5
1.2 Objetivo de la carrera	6
<b>Capítulo II: Aspectos específicos del módulo</b>	
2.1 Presentación	7
2.2 Propósito del módulo	9
2.3 Mapa del módulo	10
2.4 Unidades de aprendizaje	11
2.5 Referencias	22

## CAPÍTULO I: Generalidades del Profesional Técnico-Bachiller

### 1.1 Marco Curricular Común de la Educación Media Superior

El Marco Curricular Común de la Educación Media Superior propone una apuesta curricular centrada en el desarrollo integral de las y los adolescentes y jóvenes, con la finalidad de formar estudiantes capaces de conducir su vida hacia su futuro con bienestar y satisfacción; con sentido de pertenencia social, conscientes de los problemas sociales, económicos y políticos que aquejan al país, dispuestos a participar de manera responsable y con toma de decisión hacia los procesos de la democracia participativa y compromiso por generar soluciones de las problemáticas que los aquejan y que tengan la capacidad de aprender a aprender en el trayecto de su vida. Que sean adolescentes y jóvenes capaces de erigirse como agentes de transformación social y que fomenten una cultura de paz y de respeto hacia la diversidad social, sexual, política y étnica; solidarios y empáticos.

A través del currículum laboral, el Profesional Técnico-Bachiller desarrollará competencias laborales extendidas pertinentes, buscando la transversalidad con los módulos del currículum fundamental y ampliado; permitiendo con ello desarrollar conocimientos, destrezas, habilidades, actitudes y valores que le permitan comprender los procesos productivos en los que está involucrado para enriquecerlos, transformarlos, resolver problemas, ejercer la toma de decisiones y desempeñarse en diferentes ambientes laborales, con una actitud creadora, crítica, responsable y propositiva; de la misma manera, fomenta el trabajo en equipo, colaborativo, el desarrollo pleno de su potencial en los ámbitos profesional, personal, así como la convivencia de manera armónica con el medio ambiente y la sociedad.

## 1.2 Objetivo de la carrera

### **PT-B en Soporte y mantenimiento de equipo de cómputo.**

Realizar los servicios de instalación, configuración, operación, mantenimiento y actualización de equipo, dispositivos periféricos, sistemas y redes de computadoras, incorporando tecnologías de vanguardia.

## CAPÍTULO II: Aspectos específicos del módulo

### 2.1 Presentación

El módulo de **Programación básica orientada a objetos**, y se imparte en el quinto semestre de la carrera de Profesional Técnico-Bachiller en **Soporte y mantenimiento de equipo de cómputo**, como parte del Trayecto técnico Programación aplicada. Tiene como finalidad que la o el alumno adquiera las habilidades y destrezas necesarias para la programación orientada a objetos, utilizando las estructuras básicas para solucionar problemas cotidianos que enriquezcan la formación del técnico en su quehacer profesional.

En la actualidad la programación orientada a objetos (POO) es una de las técnicas más modernas de desarrollo que trata de disminuir el coste del software, aumentando la eficiencia y reduciendo el tiempo de espera para la puesta en práctica de una nueva aplicación. Por eso, donde la POO toma verdadera ventaja es en poder compartir y reutilizar el código. Sin embargo, no debe pensarse que esta forma de programación resuelve todos los problemas de una forma sencilla y rápida. Para conseguir buenos resultados, es preciso dedicar un tiempo mayor al análisis y al diseño; pero no será un tiempo perdido, ya que redundará en el menor tiempo empleado en la realización de aplicaciones futuras. Por ello, resulta indispensable que el técnico que realiza actividades de programación la conozca a detalle. Debido a su enfoque eminentemente práctico, ha influido en la industria del desarrollo de software, por lo que el desarrollo de competencias en esta disciplina contribuye a realizar programas altamente confiables y de ágil mantenimiento.

El presente módulo está conformado por dos unidades de aprendizaje. En la primera unidad se enfoca al desarrollo de competencias en el Manejo de componentes y estructuras básicas de programación orientada a objetos, con la finalidad de concebir y organizar los programas, que incrementarán la productividad en el desarrollo de soluciones automatizadas; en la segunda unidad, está dirigida a la Creación de programas orientados a objetos por medio de la aplicación de modelos y tecnologías basadas en objetos, de manera que el estudiante desarrolle su competencia en el uso de éstos e identifique los beneficios que le ofrecen estas herramientas de programación.

La contribución del módulo es desarrollar competencias profesionales esenciales para su perfil de egreso y para su inserción laboral, incluyendo conocimientos, destrezas, habilidades, actitudes y valores que se integran y relacionan con otros módulos del plan de estudios, para Identificar los fundamentos de la programación orientada a objetos, considerando la estructura de programas bajo este ambiente, así como el manejo de los principios de la programación orientada a objetos. El manejo de los modelos de la programación orientada a objetos en el análisis, diseño y desarrollo de programas y el desarrollo de programas de aplicación final, empleando los fundamentos y características de un lenguaje Orientado a Objetos para dar solución a problemas de aplicación real.

La tarea educativa tendrá que diversificarse, a fin de que el personal docente realice funciones preceptoras, que consistirán en la guía y acompañamiento del alumnado durante su proceso de formación académica y personal y en la definición de estrategias de participación que le permitan incorporar a su familia en un esquema de corresponsabilidad que coadyuve a su desarrollo integral; por tal motivo, deberá destinar tiempo dentro de cada unidad para brindar este apoyo a la labor educativa de acuerdo con el Programa de Preceptorías.

Por otro lado, el alumnado deberá gestionar su aprendizaje, a fin de distribuir su tiempo para dedicar un porcentaje de la duración del módulo al estudio independiente, para reforzar el conocimiento previo o adquirido en clase, de tal forma que obtengan hábitos de estudio que le permitan ser autodidacta.

Finalmente, es necesario que al concluir cada resultado de aprendizaje se considere una sesión de clase en la cual se realice la recapitulación de los aprendizajes logrados, con el propósito de verificar que éstos se han alcanzado o, en caso contrario, determinar las acciones de mejora pertinentes. Cabe señalar que en esta sesión el alumno o la alumna que haya obtenido insuficiencia en sus actividades de evaluación o desee mejorar su resultado, tendrá la oportunidad de entregar nuevas evidencias.

## 2.2 Propósito del módulo

**Realizar la programación orientada a objetos, utilizando las estructuras básicas para solucionar problemas de la vida cotidiana.**

## 2.3 Mapa del módulo

Nombre del módulo	Unidad de aprendizaje	Resultado de aprendizaje
<p><b>Programación básica orientada a objetos.</b></p> <p><b>90 horas</b></p>	<p>1. Manejo de componentes y estructuras básicas de programación orientada a objetos.</p> <p><b>50 horas</b></p>	<p><b>1.1</b> Identifica los fundamentos de la programación orientada a objetos, considerando la estructura de programas bajo este ambiente.</p> <p><b>20 horas</b></p>
		<p><b>1.2</b> Maneja los principios de la programación orientada a objetos</p> <p><b>30 horas</b></p>
	<p>2. Creación de programas orientados a objetos</p> <p><b>40 horas</b></p>	<p><b>2.1</b> Maneja el paradigma de la programación orientada a objetos en el análisis, diseño y desarrollo de programas.</p> <p><b>20 horas</b></p>
		<p><b>2.2</b> Desarrolla programas de aplicación final, empleando los fundamentos y características de un lenguaje orientado a objetos para dar solución a problemas de aplicación real.</p> <p><b>20 horas</b></p>

## 2.4 Unidades de aprendizaje

<b>Unidad de aprendizaje:</b>	<b>1. Manejo de componentes y estructuras básicas de programación orientada a objetos</b>	<b>50 horas</b>
<b>Propósito de la unidad</b>	Manejar los componentes y estructuras básicas de la programación orientada a objetos para resolver problemas cotidianos de manera eficiente	
<b>Resultado de aprendizaje:</b>	<b>1.1</b> Identifica los fundamentos de la programación orientada a objetos, considerando la estructura de programas bajo este ambiente.	<b>20 horas</b>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
<p><b>1.1.1</b> Define un ejemplo por escrito de la vida cotidiana, en donde se identifiquen los fundamentos de la Programación orientada a objetos.</p>	<ul style="list-style-type: none"> <li>Ejemplo práctico del entorno: un cuento, un videojuego, una biblioteca, un árbol genealógico, etc.</li> </ul>	<p><b>10%</b></p>	<p><b>A.</b> Introducción a la programación orientada a objetos.</p> <ul style="list-style-type: none"> <li>Paradigma de la Programación orientada a objetos (Origen)</li> <li>Concepto de programación.</li> <li>Tipos de programación.                             <ul style="list-style-type: none"> <li>- Orientada a procedimientos</li> <li>- Estructurada</li> <li>- Basada en objetos</li> <li>- Orientada a objetos.</li> </ul> </li> <li>Algoritmo</li> <li>Diagramas de flujo.</li> <li>Pseudocódigo</li> <li>Codificación</li> <li>Generalidades de una estructura de programa.                             <ul style="list-style-type: none"> <li>- Sintaxis</li> <li>- Codificación</li> <li>- Compilación.</li> <li>- Detección de errores</li> </ul> </li> <li>Objeto                             <ul style="list-style-type: none"> <li>- Estado (Atributo)</li> <li>- Comportamiento (Método)</li> <li>- Identidad</li> </ul> </li> <li>Asociación de objetos con situaciones reales.</li> </ul>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
			<ul style="list-style-type: none"> <li>• Características de la Programación Orientada a Objetos                             <ul style="list-style-type: none"> <li>- Abstracción</li> <li>- Encapsulamiento</li> <li>- Modularidad</li> <li>- Principio de ocultación</li> <li>- Polimorfismo</li> <li>- Herencia</li> <li>- Recolector de basura</li> </ul> </li> <li>• Ventajas y desventajas de los programas orientados a objetos.</li> </ul> <p><b>B. Identificación de los Conceptos fundamentales de la Programación Orientada a Objetos</b></p> <ul style="list-style-type: none"> <li>• Objeto                             <ul style="list-style-type: none"> <li>- Estructura</li> <li>- Organización</li> <li>- Identificación</li> </ul> </li> <li>• Clase                             <ul style="list-style-type: none"> <li>- Subclase</li> </ul> </li> <li>• Herencia (Tipos)                             <ul style="list-style-type: none"> <li>- Herencia Simple</li> <li>- Herencia múltiple</li> </ul> </li> </ul> <p><b>C. Características de los:</b></p> <ul style="list-style-type: none"> <li>• Lenguajes orientados a objetos                             <ul style="list-style-type: none"> <li>- Herencia Múltiple</li> <li>- Eficiencia</li> <li>- Asignación de tipos</li> <li>- Manejo de Memoria</li> <li>- Encapsulación</li> </ul> </li> <li>• Lenguajes de programación orientados a objetos</li> </ul>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
			<ul style="list-style-type: none"> <li>- C++</li> <li>- C#</li> <li>- Object Pascal (Delphi)</li> <li>- Java</li> <li>- JavaScript</li> <li>- Objective-C</li> <li>- PHP (a partir de su versión 5)</li> <li>- Python</li> <li>- Ruby</li> <li>- VB.NET</li> <li>- Visual FoxPro (versión 6)</li> <li>- Visual Basic 6.0</li> </ul>
<p><b>Sesión para recapitulación y entrega de evidencias, al término de cada resultado de aprendizaje.</b></p>			

Resultado de aprendizaje:	1.2 Maneja los principios de la programación orientada a objetos.		30 horas
Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
<p>1.2.1 Elabora un programa básico describiendo los componentes y características de la programación orientada a objetos.</p>	<ul style="list-style-type: none"> <li>Programa básico con la descripción de los componentes y características de la programación orientada a objetos.</li> </ul>	<p>20 %</p>	<p>A. Representación de la programación orientada a objetos en las computadoras</p> <ul style="list-style-type: none"> <li>Objetos en la programación orientada a objetos</li> <li>Clases                             <ul style="list-style-type: none"> <li>- Declaración de una clase</li> <li>- Instanciación de una clase en un objeto</li> <li>- Operación de un objeto</li> <li>- Atributos dinámicos en objetos</li> </ul> </li> <li>Relaciones entre objetos                             <ul style="list-style-type: none"> <li>- Composición</li> <li>- Uso</li> <li>- Delegación</li> </ul> </li> <li>Tipos de datos básicos                             <ul style="list-style-type: none"> <li>- Char</li> <li>- Int</li> <li>- Float</li> <li>- Doble</li> <li>- Void</li> </ul> </li> <li>Variables                             <ul style="list-style-type: none"> <li>- Nombre, declaración y uso</li> <li>- Tipos de variables</li> </ul> </li> <li>Locales</li> <li>Globales</li> <li>De parámetro                             <ul style="list-style-type: none"> <li>- Variables estáticas locales</li> <li>- Variables estáticas globales</li> </ul> </li> <li>Constantes</li> <li>Operadores                             <ul style="list-style-type: none"> <li>- Aritméticos</li> </ul> </li> </ul>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
			<ul style="list-style-type: none"> <li>- Relacionales</li> <li>- Lógicos</li> <li>• Punteros y referencias</li> </ul> <p><b>B. Estructura de un programa en un Lenguaje Orientado a Objetos</b></p> <ul style="list-style-type: none"> <li>• Comentarios</li> <li>• Función principal ()</li> <li>• Archivos de cabecera</li> <li>• Cuerpo del programa</li> <li>• Entrada y salida de datos</li> <li>• Sentencias de control</li> </ul> <p><b>C. Manejo de las clases en la programación.</b></p> <ul style="list-style-type: none"> <li>• Conceptos que representan.                             <ul style="list-style-type: none"> <li>- Variables.</li> <li>- Datos.</li> </ul> </li> <li>• Componentes de la clase.                             <ul style="list-style-type: none"> <li>- Atributos</li> <li>- Métodos.</li> </ul> </li> <li>• Atributos.                             <ul style="list-style-type: none"> <li>- Atributos pasivos.</li> <li>- Variables miembros.</li> <li>- Campos.</li> </ul> </li> <li>• Métodos.                             <ul style="list-style-type: none"> <li>- Funciones miembros</li> <li>- Atributos activos.</li> <li>- Operaciones.</li> <li>- Comportamiento.</li> <li>- Responsabilidades.</li> </ul> </li> <li>• Tipos de funciones miembro.                             <ul style="list-style-type: none"> <li>- Funciones (procedimientos)</li> <li>- Constructores.</li> </ul> </li> </ul>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
			<ul style="list-style-type: none"> <li>- destructores.</li> <li>• Características constructoras.                             <ul style="list-style-type: none"> <li>- Nombre.</li> <li>- Valor de retorno.</li> <li>- Número de constructores por clase.</li> <li>- Relación entre tipo y/o cantidad de parámetro.</li> <li>- Objetos de la clase invocan constructor.</li> <li>- compilador genera constructor por defecto.</li> <li>- Características de constructor por defecto.</li> <li>- Suministro de parámetros a su invocación.</li> <li>- Sobrecargas.</li> </ul> </li> <li>• Características destructoras.</li> <li>• Variables de clases y su relación con:                             <ul style="list-style-type: none"> <li>- Atributos</li> <li>- Clase e instancias</li> <li>- Copias almacenadas.</li> <li>- Independencia con los objetos de clase.</li> <li>- Definida como static.</li> </ul> </li> </ul>
<p><b>Sesión para recapitulación y entrega de evidencias, al término de cada resultado de aprendizaje.</b></p>			

<b>Unidad de aprendizaje:</b>	<b>2. Creación de programas orientados a objetos</b>	<b>40 horas</b>
<b>Propósito de la unidad</b>	Desarrollar programas, utilizando los fundamentos, estructuras, modelos de diseño, diagramas y técnicas básicas de la programación orientada a objetos, para dar solución a problemas.	
<b>Resultado de aprendizaje:</b>	<b>2.1</b> Maneja el paradigma de la programación orientada a objetos en el análisis, diseño y desarrollo de programas.	<b>20 horas</b>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
<p><b>2.1.1</b> Desarrolla un programa en lenguaje C++, manejando las propiedades de objetos y clases</p>	<ul style="list-style-type: none"> <li>Programa desarrollado, compilado y ejecutado.</li> </ul>	<p><b>35 %</b></p>	<p><b>A.</b> Manejo de objetos con un lenguaje de programación.</p> <ul style="list-style-type: none"> <li>Lenguaje C++.</li> <li>Librerías para C++.</li> <li>Sintaxis C++.</li> <li>Operadores.</li> <li>Diseño de objetos</li> <li>Creación y destrucción de objetos.</li> <li>Mensajes entre objetos.</li> <li>Interfases.</li> <li>Encapsulamiento.</li> <li>Genericidad.</li> <li>Sintaxis lenguaje C++.                             <ul style="list-style-type: none"> <li>- Uso de &lt;nombre&gt;.h</li> <li>- Atributos</li> <li>- Variables.</li> <li>- Funciones miembro.</li> <li>- Método, uso de &lt;nombre&gt;.ccp</li> <li>- Constructor.</li> <li>- destructores.</li> </ul> </li> </ul> <p><b>B.</b> Manejo de las propiedades de la programación orientada a objetos en situaciones reales:</p> <ul style="list-style-type: none"> <li>Herencia</li> <li>Polimorfismo</li> <li>Aplicación en un programa</li> </ul>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
			<ul style="list-style-type: none"> <li>• Aplicación en el lenguaje orientado a objetos C++.</li> </ul> <p><b>C. Manejo del análisis orientado a objetos.</b></p> <ul style="list-style-type: none"> <li>• Comprender el problema.</li> <li>• Elaborar las especificaciones de requerimientos del usuario y del software.</li> <li>• Identificar los objetos y sus atributos.</li> <li>• Identificar los servicios que deberá ofrecer cada objeto (interfase).</li> <li>• Establecer interconexiones (colaboraciones) entre los objetos, en términos de servicios.</li> </ul> <p><b>D. Uso del compilador del lenguaje de programación</b></p> <ul style="list-style-type: none"> <li>• Codificación del programa.</li> <li>• Compilación del programa fuente.</li> <li>• Depuración de los posibles errores aparecidos.</li> <li>• Linkear o enlazar los componentes del programa objeto.</li> <li>• Uso de teclas rápidas en el compilador:</li> <li>• Ejecución del programa</li> </ul> <p><b>E. Manejo de la metodología UML de representación de modelos, en el modelaje y desarrollo de programas (diagramas).</b></p> <ul style="list-style-type: none"> <li>• Objetivo de la metodología.             <ul style="list-style-type: none"> <li>- Visualizar.</li> <li>- Especificar.</li> <li>- Construir.</li> <li>- Documentar.</li> </ul> </li> <li>• Clases de bloques en la representación de modelos.</li> </ul>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
			<ul style="list-style-type: none"> <li>- Elementos.</li> <li>- Relaciones.</li> <li>- Diagramas.</li> <li>• Diagrama de clases.</li> <li>• Diagrama de objetos.</li> <li>• Maneja los pasos en el diseño orientado a objetos.</li> <li>• Revisar objetos creados en etapa de análisis</li> <li>• Especificar dependencias entre clases.</li> <li>• Organizar las jerarquías de clases.</li> <li>• Diseñar las clases</li> <li>• Diseñar funciones miembro.</li> <li>• Diseñar el programa principal.</li> </ul>
<p><b>Sesión para recapitulación y entrega de evidencias, al término de cada resultado de aprendizaje.</b></p>			

<b>Resultado de aprendizaje:</b>	<b>2.2</b> Desarrolla programas de aplicación final, empleando los fundamentos y características de un lenguaje orientados a objetos para dar solución a problemas de aplicación real.	<b>20 horas</b>	
Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
<p><b>2.2.1</b> Desarrolla un programa de aplicación final relevante para su entorno, empleando la metodología y técnica de un lenguaje de programación orientada a objetos.</p>	<ul style="list-style-type: none"> <li>Programa integrando componentes, estructuras, modelos de diseño y representación de la programación orientada a objetos, compilado y en ejecución</li> </ul>	<p><b>35 %</b></p>	<p><b>A.</b> Desarrollo de un programa orientado a objetos, relacionando la metodología con el lenguaje C++.</p> <ul style="list-style-type: none"> <li>Funciones de librería estándar</li> <li>Implementación de una clase.</li> <li>Constructores y destructores</li> <li>Anidamiento de funciones miembro.</li> <li>Funciones miembro privadas.</li> <li>Arrays dentro de una clase.</li> <li>Reserva de memoria para objetos.</li> <li>Miembro de clase estático.</li> <li>Función miembro estática.</li> <li>Arrays de objetos.</li> <li>Tipos de datos.</li> <li>Tipo struct, y class.</li> <li>Objetos como argumentos de funciones</li> <li>Funciones amigas.</li> <li>Intercambio de datos privados de clases.</li> <li>Sobrecarga de operadores                         <ul style="list-style-type: none"> <li>Devolución de objetos.</li> </ul> </li> <li>Funciones miembros const.</li> </ul> <p><b>B.</b> Proceso específico para desarrollar un programa en Lenguaje C++</p> <ul style="list-style-type: none"> <li>Sinopsis</li> <li>Presentación del problema</li> <li>Comprender el problema                         <ul style="list-style-type: none"> <li>Análisis</li> <li>Especificación</li> <li>Límites</li> </ul> </li> </ul>

Actividades de evaluación	Evidencias por recopilar	Ponderación	Contenidos
			<ul style="list-style-type: none"> <li>• Diseñar los ficheros y módulos                             <ul style="list-style-type: none"> <li>- Definición de módulos y librerías</li> </ul> </li> <li>• Escribir el código</li> <li>• Preproceso                             <ul style="list-style-type: none"> <li>- Preprocesador</li> <li>- Directivas de preprocesado</li> <li>- Unidad de compilación</li> </ul> </li> <li>• Generador de código                             <ul style="list-style-type: none"> <li>- Fichero .obj</li> <li>- Depurador</li> </ul> </li> <li>• Enlazado                             <ul style="list-style-type: none"> <li>- Versión de depuración</li> <li>- Versión de publicación</li> <li>- Librerías</li> </ul> </li> <li>• Errores                             <ul style="list-style-type: none"> <li>- Tipos de errores</li> <li>- Gravedad de los errores</li> <li>- Errores de ejecución</li> </ul> </li> <li>• Recursos</li> </ul>
<p><b>Sesión para recapitulación y entrega de evidencias, al término de cada resultado de aprendizaje.</b></p>			

## 2.5 Referencias

### Básicas:

- Balagurusamy, E. (2006). *Programación orientada a objetos con C++*. Mc Graw Hill.
- Fernández Sastre, S. (2006) *Fundamentos de diseño y programación orientada a objetos*. Mc Graw Hill.
- Ceballos S., Francisco J. (2018). *Programación Orientada a Objetos con C+*. 5a Ed. Ra-Ma, Librería y Editorial Microinformática.
- Programación Fácil. (2020). *El MÉTODO \_\_init\_\_ y SELF - Curso desde cero Programación Orientada a Objetos Python - Capítulo 3* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=ID518qg86qg>
- Lucid, Software Español. (2019, 4 febrero). *Tutorial - Diagrama de clases UML* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=Z0yLerU0g-Q>
- Workana, & Workana. (2021, 5 abril). *C++: Qué es, Para qué sirve, Ventajas y Desventajas*. Glosario - Workana | el Glosario Workana Explica Terminología del Mundo Freelance, Conceptos Fundamentales del Marketing y los Negocios. <https://i.workana.com/glosario/que-es-c/>

### Complementarias:

- Sergio M, Fernández, S. *Fundamentos del diseño y la programación orientada a objetos*. Editorial McGraw Hill.