

# Modelo de enseñanza para la materia de programación orientada a objetos basado en rutas de aprendizaje y objetos de aprendizaje.

## Learning model for object-oriented programming topic based on learning paths and learning objects.

Amalia Elvira Jiménez López (1).

Estudiante Tecnológico Nacional de México/Instituto Tecnológico de Tuxtla Gutiérrez.

[L17271328@tuxtla.tecnm.mx](mailto:L17271328@tuxtla.tecnm.mx).

María Elena Jiménez Velasco (2). Estudiante Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez.

[L17271329@tuxtla.tecnm.mx](mailto:L17271329@tuxtla.tecnm.mx).

Héctor Guerra Crespo\* (3), Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez, [hector.gc@tuxtla.tecnm.mx](mailto:hector.gc@tuxtla.tecnm.mx).

Jorge Octavio Guzmán Sánchez (4), Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez,

[jorge.gs1@tuxtla.tecnm.mx](mailto:jorge.gs1@tuxtla.tecnm.mx).

Galdino Belizario Nango Solís (5), Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez,

[galdino.ns@tuxtla.tecnm.mx](mailto:galdino.ns@tuxtla.tecnm.mx).

Joel Gómez Pérez (6), Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez, [joel.gp@tuxtla.tecnm.mx](mailto:joel.gp@tuxtla.tecnm.mx).

---

\*corresponding author.

**Artículo recibido en octubre 28, 2022; aceptado en noviembre 30, 2022.**

### Resumen.

*Este artículo presenta un modelo de enseñanza de basado en objetos de aprendizaje principalmente basado en videos de producción audiovisual robusta para la materia de programación orientada a objetos. El modelo, en esta etapa, considera dos rutas de aprendizaje, una basada en conocimientos previos de programación y otra considera que el estudiante no tiene conocimientos en programación, estas rutas le dan versatilidad al modelo para adaptarse a diferentes planes de estudios formales a nivel universitario. La implementación del modelo es a través de un gestor de contenido educativo como Moodle y considera material didáctico en formato de apuntes, programas de ejemplo, presentaciones y videografía que permiten utilizar medios de educación a distancia y en línea.*

**Palabras claves:** Modelos educativos, objetos de aprendizaje, orientado a objetos, programación.

### Abstract.

*This paper presents a learning model based on learning objects mainly based on videos of robust audiovisual production for object-oriented programming topic. The model, at this stage, considers two learning paths, one based on prior programming knowledge and other considers that the student has no programming knowledge. These paths give the model versatility to adapt to different formal study plans at the university level. The implementation of the*

*model is through an educational content manager Moodle and considers didactic material in the format of notes, example programs, slideshows and video that allow distance and online education.*

**Keywords:** Educational models, learning objects, object-oriented, programming.

## 1. Introducción.

El aprendizaje no es lineal es una red compleja de puntos (nodos de conocimiento) que integrándolos generan conocimiento sólido.

La enseñanza de la materia programación orientada a objetos es un ejemplo claro de que el aprendizaje no es lineal, es una materia abstracta y es recomendable tener conocimiento previo en otras asignaturas de programación como lógica de programación, programación estructurada, estructura de datos, entre otros, sin embargo, también tiene el potencial de poder impartirse en un primer curso de programación.

Para que el tema de programación orientada a objetos pueda impartirse desde el principio de la formación de un programador será necesario enfocar los temas de las materias de conocimiento previo directo al contexto de esta filosofía de programación.

El programa de estudios típico de las áreas de programación es.

Algoritmos.

- Tipos de datos, lógica numérica (float, int).
- Operadores.
- Expresiones.
- Entrada/salida.
- Decisiones.
- Ciclos.

Programación estructurada.

- Operadores y expresiones.
- Entrada/Salida.
- Decisiones.
- Ciclos.
- Funciones.
- Arreglos.

Estructura de datos.

- Introducción.
- Arreglos.
- Búsquedas.
- Ordenamiento.
- Recursividad.
- Pilas y colas.
- Listas.
- Árboles.

Programación orientada a objetos.

- Sobrecarga de funciones.
- Sobrecarga de operadores.
- Clases.

- Constructores.
- Destrucción.
- Funciones amigables.
- Herencia.
- Composición de datos.
- Funciones virtuales.
- Plantillas.

Bajo este esquema programación orientada a objetos debe impartirse después de algoritmos y programación estructurada y de manera paralela a estructura de datos, este formato es el más común en los planes de estudios de carreras relacionadas con el tema.

## 2. Métodos.

### Programas de estudios.

La forma tradicional de impartir conocimiento inclusive una carrera es a través de un plan de estudios, el cual se define como “Conjunto estructurado de asignaturas, prácticas y actividades de la enseñanza y el aprendizaje. El plan de estudios debe contener los propósitos de formación general, los contenidos fundamentales de estudio y los criterios y procedimientos de evaluación y acreditación” (INEGI 2011, p. 7) y el detalle de cada asignatura o curso queda plasmado en un programa de estudios el cual se define como “La descripción sintetizada de los contenidos de las asignaturas o unidades de aprendizaje, ordenadas por secuencias o por áreas relacionadas con los recursos didácticos y bibliográficos indispensables, con los cuales se regulará el proceso de enseñanza-aprendizaje” (INEGI 2011, p. 7).

Los programas de estudio actuales para la impartición de programación orientada a objetos están consolidados dentro de un plan de estudios global, consideran que esta asignatura se encuentra dentro del área de programación de propósito general junto con las materias algoritmos, programación estructurada, programación avanzada, estructura de datos, entre otros, por lo tanto, para el desarrollo del programa de estudios de programación orientada a objetos hay que considerar todas las asignaturas mencionadas. La propuesta de este proyecto es aislar la programación orientada a objetos en un sólo programa de estudio que permita la flexibilidad de importación en cualquier momento del plan de estudios y con flexibilidad en el tiempo para impartirse.

### Objetos de aprendizaje.

La propuesta de este proyecto en cuanto al desarrollo del material didáctico es a través de objetos de aprendizaje definido como “objeto de aprendizaje es cualquier entidad -digital o no- que puede ser usada para aprender, enseñar o capacitar (training)” (Willey, 2000). Otras definiciones incorporan “soportado por tecnología” (Willey, 2000) y evaluación (L’Allier, 1998).

De acuerdo a (South y Monson, 2000) de acuerdo al estándar (IEEE Std 1484.12.1, 2002) los atributos de Objetos Aprendizaje son.

- Durabilidad.
- Interoperabilidad.
- Accesibilidad.
- Reusabilidad.
- Extensibilidad.
- Asequibilidad.
- Manejabilidad.

Los recursos considerados para el desarrollo de los objetos de aprendizaje son.

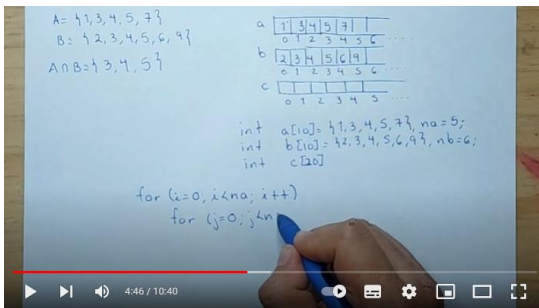
- Pizarrón blanco.
- Presentación estática.
- Grabación de pantalla.
- Cuaderno, mano alzada.
- Elementos de edición.



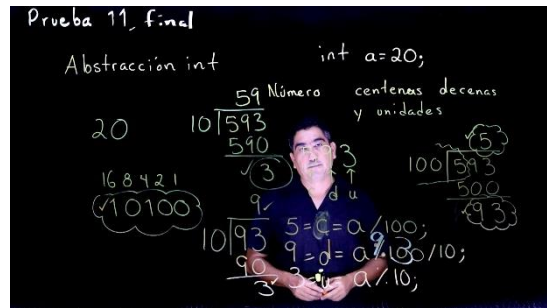
(a)



(b)



(c)



(d)

**Figura 1.** Recursos de aprendizaje (a) pizarrón, (b) presentación y grabación de pantalla, (c) cuaderno-mano alzada, (d) elementos edición pizarrón de cristal.

### 3. Desarrollo.

El modelo de enseñanza de la materia de programación orientada a objeto propuesta está preparado para impartirla desde cero, sin antecedentes de programación o una vez llevada la materia de programación estructurada considerando que dentro de los temas se tengan contemplado funciones y ciclos al menos. En la figura 2 se presenta el modelo.

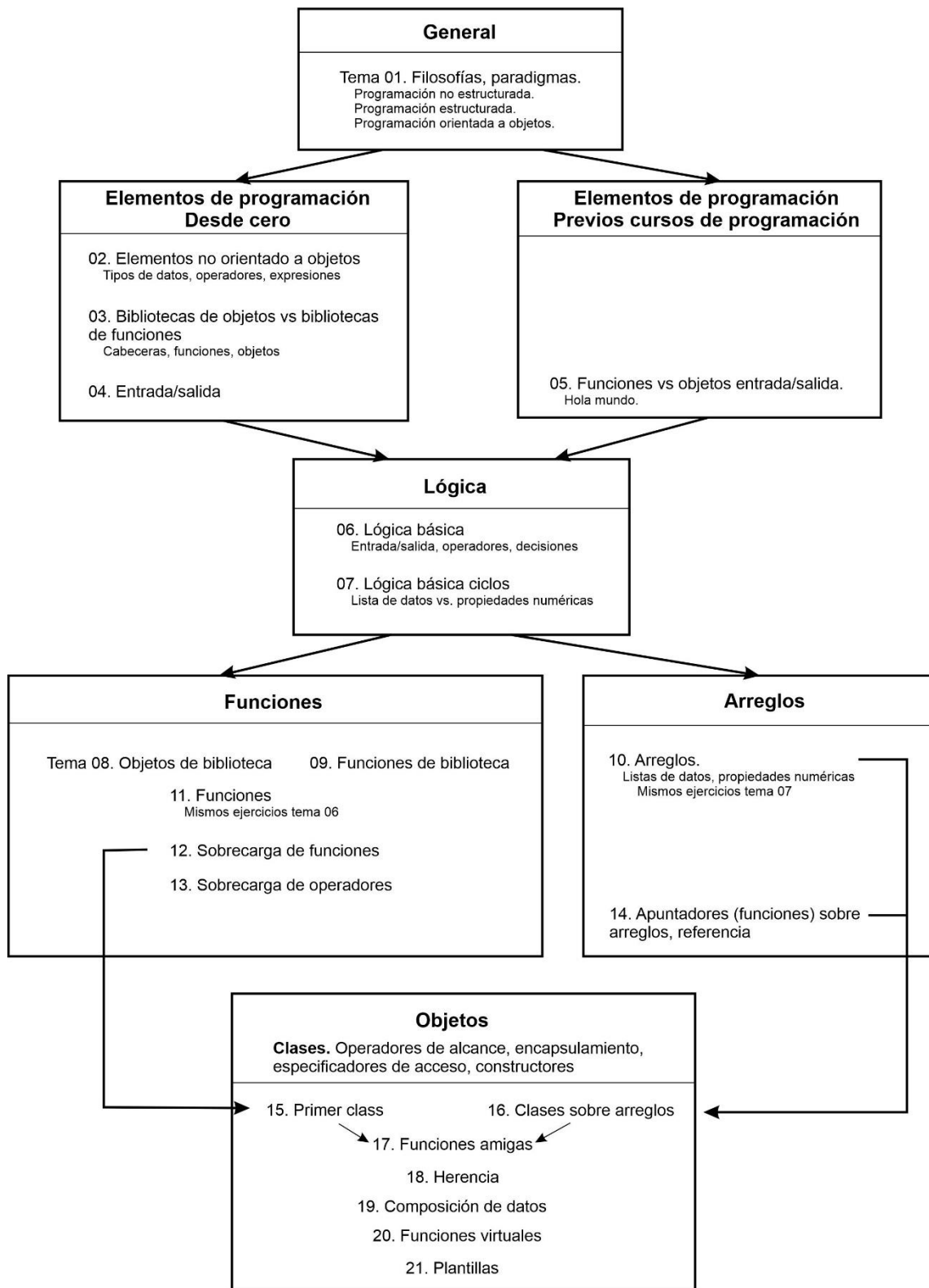


Figura 2. Modelo de enseñanza de programación orientada a objetos.

El modelo consta de 21 temas que descritos linealmente son.

- Tema 01. Filosofías, paradigmas.
- Tema 02. Elementos no orientados a objetos.
- Tema 03. Bibliotecas de objetos vs. bibliotecas de funciones.
- Tema 04. Entrada/salida.
- Tema 05. Funciones vs. Objetos (entrada/salida).
- Tema 06. Lógica básica, decisiones.
- Tema 07. Lógica básica, ciclos.
- Tema 08. Objetos de biblioteca.
- Tema 09. Funciones de biblioteca.
- Tema 10. Arreglos.
- Tema 11. Funciones.
- Tema 12. Sobrecarga de funciones.
- Tema 13. Sobrecarga de operadores.
- Tema 14. Apuntadores.
- Tema 15. Primer class.
- Tema 16. Clases sobre arreglos.
- Tema 17. Funciones amigas.
- Tema 18. Herencia.
- Tema 19. Composición de datos.
- Tema 20. Funciones virtuales.
- Tema 21. Plantillas.

Se utiliza el término temas para no dirigirlo formalmente a un tipo de plan de estudio en particular y pueda adaptarse a objetivos, unidades o competencias.

Consta de 7 secciones.

1. General.
2. Elementos de programación, desde cero.
3. Elementos de programación, previos cursos de programación.
4. Lógica.
5. Funciones.
6. Arreglos.
7. Objetos.

El modelo tiene dos rutas de aprendizaje, una ruta considera empezar sin conocimientos previos podría decirse que para primer semestre y la otra considera que el estudiante tiene lógica de programación sabe programar con funciones y arreglos, en esta ruta podría impartirse el modelo desde tercer semestre.

El tiempo para impartir los temas a detalle por lo pronto no están definidos en el modelo, pero ambas rutas consideran el mismo tiempo para impartirse, en el mismo semestre en un rango de 60 a 90 horas, con la observación que la ruta sin conocimientos previos no contempla el tema de arreglos, por lo tanto, el material didáctico en la parte de programación orientada a objetos en esa ruta debe considerarlo.

### **Implementación.**

La implementación del modelo hasta el momento es en la plataforma Moodle la cual permite la gestión de contenido educativo y el manejo de la ruta de aprendizaje es a discreción por parte del docente y el estudiante, existe potencial para desarrollar una aplicación que automatice el proceso y debe ser considerada como un siguiente paso en el desarrollo del modelo.



Figura 3. Plataforma Moodle I.T.T.G. Curso programación orientada a objetos, organización de pestañas.

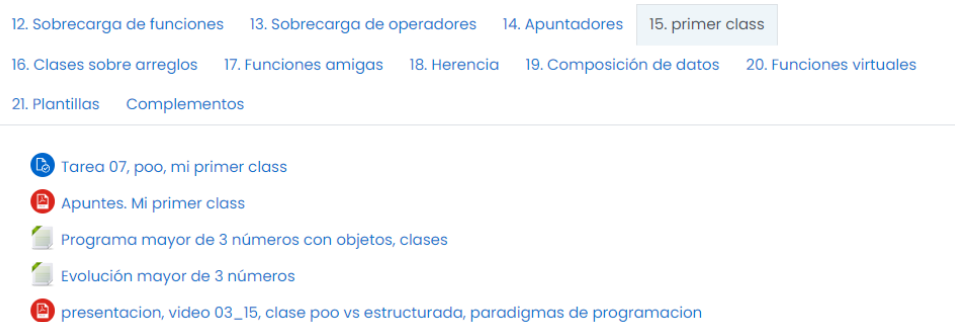


Figura 4. Estructura del contenido por pestaña, tarea, apuntes, programas y presentaciones. Pestaña 15. Primer class.

En cada pestaña el documento eje es la tarea la cual define el flujo de aprendizaje empezando por la definición del problema a resolver como lo ilustra la figura 5.

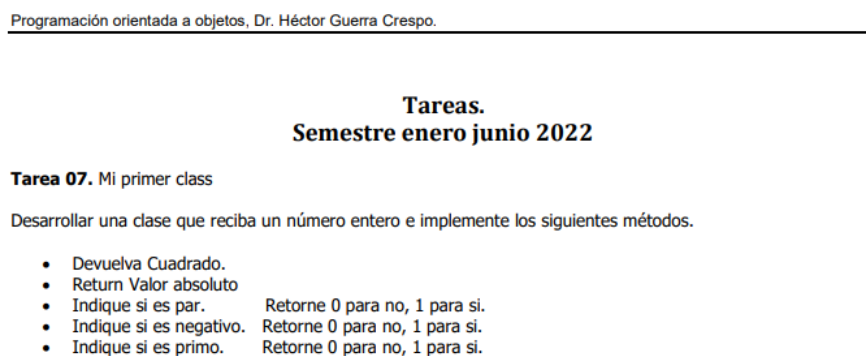


Figura 5. Documento de la tarea, definición de la tarea.

Continua con la definición de la fecha de entrega, el formato de entrega y las actividades a realizar como lo ilustra la figura 6. Posteriormente se define la videografía (objetos de aprendizaje, videos) necesaria para desarrollar la actividad, gracias a los elementos didácticos y objetos de aprendizaje el modelo puede implementarse en modalidad a distancia y en línea.

Fecha de entrega: **viernes 01 de abril.**

Entrega.

- Documento [pdf](#) con los códigos fuente y las salidas.

Para realizar la tarea realizar las siguientes actividades.

Consultar el video 03\_14 del capítulo 3 en la plataforma <https://tecnodidactica.com>

Leer los archivos.

- Apuntes clases ([class](#)) [poo](#)
- Evolución mayor de 3 números
- Programa mayor de 3 números con objetos, clases
- Presentación video 03\_15, clase [poo](#) vs estructurada, paradigmas de programación

Ver los videos de [youtube](#)

<https://youtu.be/AaSYJU4qYgs>

clase 01a, evolución [struct](#) a [class](#), conceptos básicos

<https://youtu.be/w03qB0cYbzY>

clase 01b, evolución [struct](#) a [class](#), conceptos básicos

[https://youtu.be/vXM9I9Iu\\_dw](https://youtu.be/vXM9I9Iu_dw)

clase 01c, evolución [struct](#) a [class](#), conceptos básicos

<https://youtu.be/cHGhRgFl5sw>

clase 01d, evolución [struct](#) a [class](#), conceptos básicos

<https://youtu.be/XekcZxt17to>

clase 01e, evolución [struct](#) a [class](#), conceptos básicos

<https://youtu.be/uD4GIgUgcgs>

clase 01f, evolución [struct](#) a [class](#), conceptos básicos

**Figura 6.** Documento de la tarea, fechas de entrega actividades y videografía correspondiente.

### Desarrollo del material didáctico.

El desarrollo del material didáctico está basado en objetos de aprendizaje audiovisuales de alto nivel de edición, cada tema se desarrolla textualmente de manera tradicional, pero se definen videos para complementar o facilitar el aprendizaje, la figura 7, 8 y 9 ilustran este aspecto.

Entonces existen `printf` y `scanf` como funciones de entrada y salida y `cout` y `cin` como objetos de entrada y salida, como usuario final el resultado es el mismo.

C# y Python son lenguajes que trabajan exclusivamente objetos, inclusive Python va más lejos ya que inclusive los tipos de datos son objetos lo cual le da un manejo muy versátil a todo el lenguaje.

Detallando más, la programación orientada a objetos no sustituye a la programación estructurada como sí la programación estructurada sustituyó la no estructurada, el código "núcleo" en la solución de un problema sigue siendo programación estructurada, la programación orientada a objetos es a nivel administración de toda la solución, lo que hace complicado entender sus diferencias.

En el siguiente video se explica la diferencia entre las diferentes filosofías de programación.



03\_14. Paradigmas de programación, C++ Builder, Turbo C++.

**Figura 7.** Desarrollo del material didáctico textual y videos relacionados, Tema 01. Filosofías, paradigmas y Tema 03. Biblioteca de objetos vs. biblioteca de funciones.



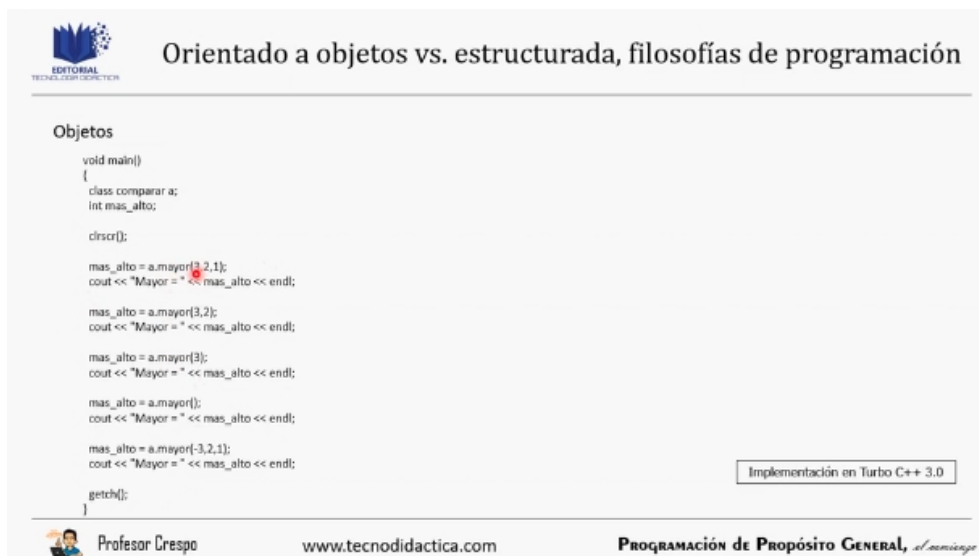


Figura 8. Video Tema 01. Filosofías, paradigmas. Objeto de aprendizaje, presentación estática y captura de pantalla, edición de múltiples videos.

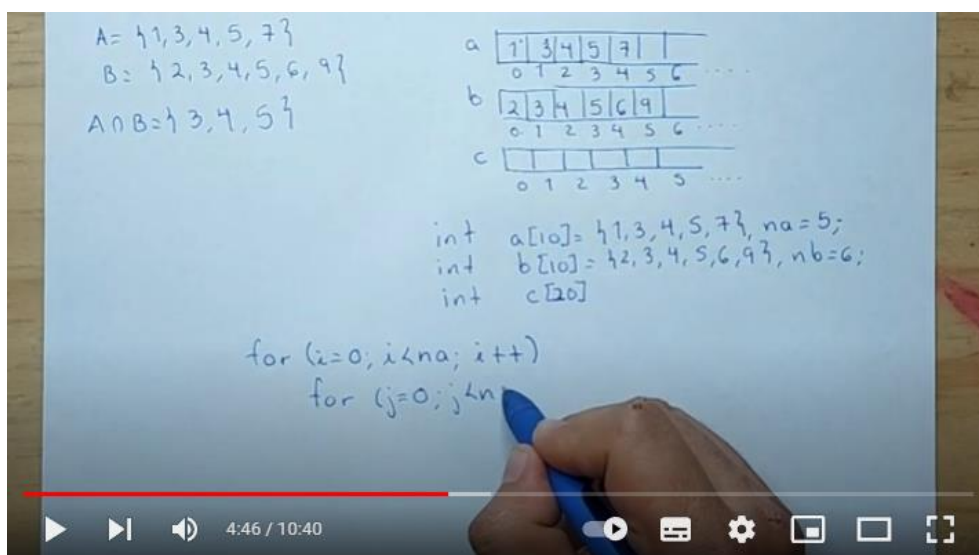


Figura 9. Video Tema 10. Arreglos. Objeto de aprendizaje, cuaderno – mano alzada.

## Conclusiones.

La pandemia covid exigió que todos los elementos para impartir educación fueran reinventados, el modelo presentado responde a esta exigencia, considera una gran parte de los temas a considerar en programación orientada a objetos y sus antecedentes académicos, es versátil porque presenta dos rutas de aprendizaje pero tiene potencial para considerar más rutas, puede adaptarse a educación formal y continua, impartirse en formato presencial, a distancia y en línea, el material didáctico propuesto integra elementos tradicionales y audiovisuales y puede servir de referencia para otros modelos.

El modelo se implementa en el gestor de contenido educativo Moodle pero lo ideal es desarrollar una aplicación web y móvil que automatice el recorrido de la rutas inclusive existe el potencial de que los usuarios definan su propia ruta de aprendizaje.

### Créditos.

Los autores agradecen al Tecnológico Nacional de México por el financiamiento del proyecto 14342.22-P (“modelo basado en objetos de aprendizaje de compleja producción audiovisual para la enseñanza de la programación orientada a objetos”), de la convocatoria de apoyo a proyectos de desarrollo tecnológico e innovación 2022 y las facilidades del Instituto Tecnológico de Tuxtla Gutiérrez para la realización de este trabajo.

### Referencias bibliográficas.

**IEEE Std 1484.12.1, 2002.** "IEEE Standard for Learning Object Metadata," in IEEE Std 1484.12.1-2002 , pp.1-40, 6 septiembre. 2002, doi: 10.1109/IEEESTD.2002.94128.

**INEGI (2012).** Clasificación mexicana de programas de estudios por campo de formación académica, Educación superior y media superior. Página 7. <https://www.copaes.org/documentos/Anexo-A-Clasificacion-Mexicana-de-Programas-de-Estudio.pdf>. Consultado el 27 de septiembre de 2022.

**L’Allier, J. J. (1998).** NETg's precision skilling: The linking of occupational skills descriptors to training interventions. Available: <http://www.netg.com/research/pskillpaper.htm>.

**South Joseph y Monson David, 2000.** “A universitywide system for creating, capturing and delivering learning objects” En: WILEY, D. A. (ed.). The instructional use of learning objects, 2000. Disponible en: <http://reusability.org/read/chapters/south.doc>. Consultado el 23 de mayo de 2022.

**Wiley D. (2000).** The instructional Use of Learning Objects. Online versión. 2000. Url. <http://www.reusability.org/read/>. Consultado el 23 de mayo de 2022

### Información de los autores.



**Amalia Elvira Jiménez López** es estudiante de la carrera de Ingeniería en Sistemas Computacionales en el Instituto Tecnológico de Tuxtla Gutiérrez se ha especializado en la edición compleja de video.



**María Elena Jiménez Velasco** es estudiante de la carrera de Ingeniería en Sistemas Computacionales en el Instituto Tecnológico de Tuxtla Gutiérrez se ha especializado en la edición de objetos de aprendizaje en Powtoon



**Héctor Guerra Crespo** es egresado del I. T. de Mérida (Yucatán, México) de la carrera de Ingeniería en Sistemas Computacionales en 1994, es Doctor en Sistemas Computacionales por la Universidad del Sur (Chiapas, México) en 2011. Es profesor en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez y en el área de Licenciatura en Sistemas Computacionales de la Universidad Autónoma de Chiapas, en ambas desde 1995. Es miembro del Claustro Doctoral "Doctorado en Ciencias de la Ingeniería" perteneciente al Programa Nacional de Posgrados de Calidad, I.T. de Tuxtla Gutiérrez desde 2016. Miembro del Sistema Nacional de Investigadores, nivel C (2024-2021).



**Jorge Octavio Guzmán Sánchez** tiene la Maestría en Ciencias de la Computación, especialidad bases de datos y sistemas de información, es Ingeniero en Sistemas Computacionales, profesional certificado por *Microsoft* en la administración de servidores con *Windows*. Ejerce la docencia desde hace más de una década, actualmente docente del Instituto Tecnológico de Tuxtla Gutiérrez.



**Galdino Belizario Nango Solís** es Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Tuxtla Gutiérrez, Chiapas, en 1996. Es Maestro en Ciencias de la Computación por el Centro de Investigación en Computación del Instituto Politécnico Nacional, en 2001. Es Doctor en Desarrollo Tecnológico por la Universidad de Ciencia y Tecnología Descartes, en 2016. Es profesor en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez, donde imparte clases en las materias de programación lógica y lenguajes y autómatas.



**Joel Gómez Pérez** es Ingeniero en Electrónica por el Instituto Tecnológico de Tuxtla Gutiérrez (ITTG), graduado en 2005; Maestro en Ciencias en Ingeniería Mecatrónica por el Instituto Tecnológico de Tuxtla Gutiérrez (ITTG), graduado en 2013. Ha colaborado con diversas universidades; entre ellas la Universidad Politécnica de Chiapas (UPCH) en el área de Ingeniería Mecatrónica, Biomédica, Energía; entre otros. Actualmente es profesor de tiempo completo en el Instituto Tecnológico de Tuxtla Gutiérrez desde 2010 y colaborador en la Universidad del Valle de México (UVM) desde 2013. Es profesor con perfil deseable PRODEP desde 2017 y jefe de laboratorio de Física en el departamento de Ciencias Básicas.