

# 50

Fecha de presentación: julio, 2022  
Fecha de aceptación: octubre, 2022  
Fecha de publicación: diciembre, 2022

## SISTEMAS DE TAREAS DOCENTES

PARA LA ENSEÑANZA Y APRENDIZAJE DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

### **SYSTEMS OF TEACHING TASKS FOR THE TEACHING AND LEARNING OF OBJECT-ORIENTED PROGRAMMING**

Juan Carlos Fonden Calzadilla<sup>1</sup>

E-Mail: [fonden1980@gmail.com](mailto:fonden1980@gmail.com)

ORCID: <https://orcid.org/0000-0001-7478-8628>

<sup>1</sup> Universidad de Ciencias Médicas de la Habana. Cuba

#### Cita sugerida (APA, séptima edición)

Fonden Calzadilla, J. C., (2022). Sistemas de tareas docentes para la enseñanza y aprendizaje de la Programación Orientada a Objetos. *Revista Universidad y Sociedad*, 14(S6), 480-491.

#### RESUMEN

Con el empleo de los métodos de investigación analítico - sintético, modelación, enfoque de sistema y la observación, desde el curso escolar 2016 – 2017, hasta hoy, se han realizado estudios al proceso de enseñanza y aprendizaje (PEA) de la Programación en las universidades José Eduardo Dos Santos” de Angola, la Universidad Tecnológica de la Habana “José Antonio Echeverría”, y en centros de educación superior en otras latitudes; constatándose la existencia de diversas dificultades al resolver problemas con el empleo del paradigma de la Programación Orientada a Objetos (POO) y como consecuencia bajos rendimientos académicos. Situación que ha corroborado la pertinencia de elaborar y proponer la aplicación de Sistemas de Tareas Docentes, las cuales se presentan en este artículo, con una adecuada relación teórica – práctica, en conformidad con los niveles de asimilación del contenido: familiarización, reproducción, aplicación y creación que posibiliten la adquisición de resultados docentes superiores en esta disciplina.

**Palabras clave:** Programación Orientada a Objetos, sistema de tareas docentes, niveles de asimilación, aprendizaje de la programación.

#### ABSTRACT

With the use of research methods analysis and synthesis, modeling, system approach and observation, from the 2016-2017 school year, until today, studies have been carried out on the teaching and learning process of Programming in the José Eduardo universities Dos Santos ”of Angola and the Technological University of Havana” José Antonio Echeverría ”, and in higher education centers in other latitudes; verifying the existence of various difficulties when solving problems with the use of the Object Oriented Programming (OOP) paradigm and as a consequence low academic performance. Situation that has corroborated the relevance of developing and proposing the application of Teaching Task Systems, which are presented in this article, with an adequate theoretical-practical relationship, in accordance with the levels of assimilation of the content: familiarization, reproduction, application and creation that enable the acquisition of superior teaching results in this discipline.

**Keywords:** Object Oriented Programming, teaching task system, assimilation levels, learning programming.

## INTRODUCCIÓN

La enseñanza y aprendizaje de la Programación, de otras disciplinas, asignaturas o ramas del saber humano, transita por diferentes niveles de asimilación del contenido, los cuales constituyen una especie de brújula o guía para la planificación del trabajo del profesor, a fin de cumplir los objetivos propuestos. La POO, en particular, implementa programas en términos de objetos en el que se incluyen conceptos de herencia, asociación entre clases, abstracción, polimorfismo, acoplamiento, ocultamiento de datos y encapsulación (Díaz & Chamorro, 2020), por lo que requiere de un estudio conceptual profundo y estrategias de aprendizajes para su aprovechamiento.

La asimilación es el resultado de captar (adquirir) un contenido, comprender algo a un cierto nivel o grado. A través de la asimilación se incorporan en un estudiante nuevas informaciones y/o experiencias a las ya existentes.

Por otra parte, se constata en la literatura consultada la existencia de cuatro niveles de asimilación, ellos son: la familiarización con el contenido, el nivel reproductivo, la aplicación del contenido y la creación como el nivel más avanzado. Esto significa que una persona en la autogestión individual o grupal del contenido u otras formas de aprendizaje, debería recorrer esos niveles.

La enseñanza y aprendizaje de la Programación se concibe a través del desarrollo de sistemas de tareas docentes, acorde con el nivel de asimilación del contenido que transite el estudiante. Así, es aconsejable que ellas recorran, gradual y ascendentemente las etapas antes expresadas.

En esta investigación una tarea docente “es la actividad concreta que realiza un estudiante, con la orientación del profesor, a fin de ser desarrollada dentro o fuera del aula, la cual conduce al educando a la adquisición de conocimientos, desarrollo de habilidades y formación de valores. Es la célula primaria y fundamental del proceso de enseñanza y aprendizaje. Es la estructura didáctica más simple, en la que el estudiante interactúa directamente”.

En las tareas docentes se revelan y materializan los objetivos que el estudiante debe lograr y para lo cual necesita haber desarrollado las habilidades necesarias y fundamentales en su progreso (Schnitzler & Núñez, 2021), a tal efecto se expresa en Martínez, Massip & Pérez (2021) que “el estudio independiente, requiere en su efectividad de la planificación, organización y destreza, pero, al unísono, al orientarlo, en la clase incluye tareas de mayor profundidad que exigen elaboración propia”.(p.10).

En tal sentido, es prudente y apropiado, que el estudiante, en su gestión independiente de los conocimientos recorra

los diferentes niveles de asimilación del contenido antes expresados.

Un enorme cúmulo de tareas docentes se pueden realizar con el empleo de las computadoras en función de la enseñanza – aprendizaje de una asignatura entre ellas: explicar un concepto; elaborar un algoritmo; resolver un problema; observar, describir, analizar y reflexionar sobre objetos y procesos; navegar dentro de redes, softwares, discos, carpetas y archivos; la búsqueda e intercambio de información y concebir, planificar, realizar y validar un proyecto informático (Fonden, 2006).

El aprendizaje de la Programación, es un proceso continuo, complejo y planificado y su éxito depende de que se proyecten, organicen, ejecuten y controlen sistemas de tareas docentes con una adecuada relación teórico-práctica, en la que se integren elementos motivacionales, afectivos, cognitivos y valorativos. La Programación por su alto nivel de complejidad y su importancia en la formación del pensamiento lógico de los estudiantes, requiere de diversas estrategias metodológicas para su mejor comprensión. Por ello, no basta con que el estudiante sea activo, hay que considerar las acciones que permiten que esta actividad sea más fructífera.

Por su importancia en la formación del pensamiento lógico, la solución de problemas y su aporte al desarrollo de la capacidad de entender conceptos y abstracciones matemáticas en los estudiantes, países como Francia, Reino Unido, Estonia, Alemania y Estados Unidos en 2014 introdujeron gradualmente la enseñanza de la programación en la enseñanza primaria. Otros países como Finlandia, Israel, Corea del Sur, Nueva Zelanda o Grecia llevan tiempo trabajando en programas pilotos, según Fonden, Stuart & Rodríguez (2018) quienes agregan que *“Por su complejidad, expertos debaten sus beneficios en la formación intelectual del estudiante, y desde qué edad es más conveniente introducirla a la vez que se requiere de una apropiada formación de los docentes”* (p.31).

El autor de esta investigación posee una experiencia de 14 años en la enseñanza y aprendizaje de la Programación, en diferentes niveles educacionales. Al respecto, en su trayectoria como docente ha podido identificar las siguientes situaciones problemáticas (Fuentes & Moo, 2017):

- a) Errónea interpretación del texto o problema a resolver. Si no está bien redactado, es poco probable que se pueda solucionar con facilidad;
- b) Desinterés por elaborar el algoritmo que conduce a la solución del problema y/o verificar su validez con valores de entrada;

- c) Intentar la codificación sin antes establecer un plan de acción y peor aún, sin interpretar el problema;
- d) Dificultad para convertir un diagrama de flujo a pseudocódigo y viceversa;
- e) Deficiente transición del paradigma de programación estructurado al de la Programación Orientada a Objetos;
- f) No prestar la debida atención a las exigencias de sintaxis del lenguaje de programación, lo cual puede producir frustraciones en el estudiante;
- g) Desfasajes en la velocidad de aprendizaje, unos muy avanzados, otros muy rezagados con errores de sintaxis, lo que obstaculiza la fluidez de la clase;
- h) Códigos de programación no optimizados;
- i) El desconocimiento previo de la lógica de programación que generalmente se aprende en los cursos introductorios de algoritmos, sin embargo, aquí los estudiantes tienen dificultades en su aprendizaje (Díaz-Leiva & Chamorro-Atalaya, 2020);
- j) Alto número de reprobados en las evaluaciones parciales y finales;
- k) Alto grado de deserción universitaria, ocasionada por su deficiente formación vocacional, la mala gestión del tiempo, la complejidad de la asignatura y la falta de recursos didácticos de los docentes (Díaz-Leiva & Chamorro-Atalaya, 2020).

Al mismo tiempo, según criterio de Pineda-Alfonso & Duarte-Piña (2020) "Diversas investigaciones realizadas en los últimos 30 años señalan que la mayoría de los profesores universitarios no tienen las habilidades necesarias para realizar adecuadamente sus tareas de enseñanza y muy poco conocimiento sobre prácticas de enseñanza efectivas". (p.98).

Las anteriores situaciones problemáticas conducen al autor a formular el siguiente problema:

¿Qué estructura didáctica debe ser empleada en el proceso de enseñanza – aprendizaje de la POO para que el mismo se desarrolle con eficacia?

Para ello se estudió el proceso de enseñanza y aprendizaje de la Programación Orientada a Objetos en las Facultades de Ingeniería Informática e Industrial en la Universidad "José Eduardo Dos Santos" de la República de Angola y la Universidad Tecnológica de la Habana "José Antonio Echeverría" a fin de identificar la estructura didáctica más apropiada al PEA de la POO.

Para el desarrollo de este artículo se sistematizaron los estudios realizados acerca del PEA de la P.O.O, en la red de redes y otras fuentes bibliográfica, y además, se analizó

el diagnóstico del Proceso de Enseñanza – Aprendizaje en la asignatura Programación Orientada a Objetos, de sus resultados docentes, indicaciones metodológicas, libros, folletos con ejercicios resueltos y guías para estudiantes.

Por lo antes expuesto, se continúa indagando para demostrar si la elaboración de sistemas de tareas docentes, con énfasis en los niveles de asimilación, contribuye a solucionar las insuficiencias detectadas, si potencia la solución de problemas complejos y la obtención de resultados académicos superiores. Seguidamente se analizarán conceptos, vías y alternativas para la elaboración del sistema de tareas.

## MATERIALES Y MÉTODOS

La investigación desarrollada es del tipo no experimental, con un alcance descriptivo-explicativo en la que se asumen los paradigmas cualitativos y cuantitativos pues se emplean procedimientos de ambos, con enfoque longitudinal. En ella se recolectan datos e informaciones en varios momentos, los cuales se analizan y relacionan entre sí. Bajo un enfoque dialéctico materialista se emplearon diversos métodos, entre ellos:

El método de la **Sistematización**, el cual permitió al autor identificar las vías para elaborar sistemas de tareas docentes, aplicables al proceso de enseñanza - aprendizaje de la Programación, teniendo en cuenta los niveles de asimilación del contenido.

Mediante el Método de **Enfoque de Sistema** se identificaron las relaciones que existen entre los cuatro niveles de asimilación del contenido y los sistemas de tareas docentes expresados.

Con el empleo del **Análisis Documental**, se investigaron los contenidos de planes de estudio de la especialidad de informática e industrial, programas aplicados, planes de clases, libros de textos y resoluciones vigentes, sus relaciones y contradicciones manifiestas, a fin de conocer el estado actual del proceso de enseñanza – aprendizaje.

La observación a clases de programación, impartidas por profesores de las Facultades de Ingeniería Informática e Industrial en la Universidad "José Eduardo Dos Santos" de la República de Angola y la Universidad Tecnológica de la Habana "José Antonio Echeverría".

La revisión bibliográfica como complemento del método análisis documental, permitió identificar indagaciones realizadas con anterioridad acerca del tema en estudio.

El cálculo porcentual de la distribución de frecuencias fue empleado en el análisis de las distribuciones que ocurren en cada indicador medido en las observaciones a clases.

## Población y muestra

Para el diagnóstico y caracterización se trabajó con una población constituida por 220 estudiantes y 12 profesores de las Facultades de Ingeniería Informática e Industrial en la Universidad “José Eduardo Dos Santos” de la República de Angola y la Universidad Tecnológica de la Habana “José Antonio Echeverría”

Para los estudiantes se tomó una muestra estratificada por conglomerado, en el que los años primero y segundo, en los cuales se imparte la POO, representaron los estratos y de ellos se seleccionan tres grupos (90 estudiantes). Un grupo de trabajadores del curso por encuentros y dos grupos del curso regular diurno.

La muestra de los profesores se tomó teniendo en cuenta su representatividad en el proceso de enseñanza – aprendizaje de la POO, en ambas especialidades, y además, su disposición a colaborar en la investigación.

## Instrumentos para la adquisición de datos e informaciones

- La guía de observación a clases, la cual permitió obtener informaciones relevantes respecto al PEA de la POO y cómo se emplearon las tareas docentes, considerando los niveles de asimilación del contenido.

- Las observaciones al proceso de evaluación de los proyectos informáticos, especialmente en sus exposiciones ante los tribunales, permitieron evidenciar la adquisición de conocimientos, desarrollo de habilidades, formación de valores, la calidad de los softwares presentados y de las respuestas a los evaluadores.

- Las reuniones metodológicas, en las que se abordan los elementos esenciales a considerar en cada clase de POO, así como las formas de evaluar el contenido y los métodos de enseñanza y aprendizaje.

- Preguntas para evaluar, finalizados las exposiciones de los proyectos, los estados de opinión de los estudiantes acerca de las clases recibidas por sus profesores.

## Las tareas docentes

Las tareas docentes, sean simples o integradoras, son estructuras didácticas, así como lo es la clase, un programa de estudio o una unidad. Para su diseño y desarrollo se requiere de una orientación que posibilite la realización de acciones concretas, siempre en correspondencia con los niveles de asimilación del conocimiento y las características individuales de los estudiantes.

La tarea integradora es aquella, en la cual se conjugan aspectos del contenido de diversas asignaturas, disciplinas o ramas del saber humano.

Las tareas complejas son aquellas que se componen de más de una subtarea. Las simples son aquellas indivisibles (Martínez, et. al, 2021). La complejidad se relaciona directamente con las habilidades que se pretende desarrollar o por el contrario, es requisito que se hayan desarrollado previamente para ejecutarla.

Lo anterior expuesto significa que el estudiante trabaja con un sistema de tareas docentes para la formación de ciertas habilidades y al mismo tiempo, una tarea necesita del desarrollo previo de habilidades específicas para su realización. Entonces se puede expresar que tareas docentes y habilidades van de la mano (Martínez, et. al, 2010).

Por otra parte, se debe destacar que un sistema es un conjunto de elementos que se relacionan entre sí y la existencia de uno es condicionante para la existencia del otro. Entonces, un sistema de tareas docentes es un conjunto de actividades concretas, relacionadas entre sí, encaminadas al PEA de una asignatura o disciplina.

## Fundamentación didáctica

Los sistemas de tareas que se presentan en este artículo se fundamentan didácticamente en las siguientes teorías:

1. Acorde con Rodríguez (2009) “Atender a los alumnos con ritmos diferentes de aprendizaje. Organizar el trabajo de clase de distinta forma, crear espacios y tiempos de formación; fomentar la enseñanza mutua y organizar el espacio en talleres. Algunos/as alumnos/as encuentran dificultades que sobrepasan las posibilidades ordinarias de diferenciación y exigen medidas excepcionales”. (p.116).

2. Un principio pedagógico expresado en Fonden (2006), citando otros autores, relacionado con la secuencia e interrelación de las asignaturas, “donde se expresa que la relación existente entre los conocimientos de una misma materia y entre los conocimientos de esta materia y los de otras, relación que se denomina intramaterias e interasignatura, respectivamente”. Aplicando este principio se debe garantizar que cada sistema de tareas esté gradualmente diseñado y dosificado, de lo simple a lo complejo, de menor a mayor complejidad en sus contenidos.

3. Se diseñan actividades cuyo contenido demanda de niveles crecientes de complejidad en las tareas y en las cuales se reflejen los niveles de asimilación. Se les enseña a los estudiantes procedimientos que les permitan encontrar lo esencial como parte del contenido que

aprenden y agrega García-Valcárcel, et. al., (2017) “Los alumnos están más motivados cuando valoran lo que están aprendiendo, cuando se implican en tareas personalmente significativas y cuando creen que el resultado del aprendizaje está bajo su control”. (p.121).

4. Que las tareas docentes, de manera gradual, con autonomía e independencia propicien la complejidad en la gestión de la actividad cognoscitiva y el desarrollo de habilidades, lo cual impactaría positivamente en los resultados de aprendizaje de los estudiantes (Rumayor, et. al, 2011).

5. Por otra parte García-Valcarcel, et. al., (2017) expresan que “La motivación y el éxito académico se derivan del hecho de proponer actividades adaptadas a los intereses de los estudiantes y que éstos asuman responsabilidades en las tareas, sintiéndose protagonistas”. (p. 128).

6. El enfoque histórico - cultural de L. S. Vigotski y los continuadores de su obra, expresado en Fonden (2006) donde se lee “(...) la independencia del desarrollo de un estudiante se logra gracias a la cooperación con otros colegas y profesores, de lo que se deriva entonces que el desarrollo humano es resultado de una perenne y solidaria cooperación entre las personas”. (p. 87).

A continuación, se exponen sistemas de tareas en correspondencia con los niveles de asimilación del contenido, las cuales contribuirán solucionar las insuficiencias detectadas y la obtención de resultados académicos superiores en cursos de ingenierías y otros escenarios donde se estudie la Programación Orientada a Objetos.

#### NIVEL DE ASIMILACIÓN: FAMILIARIZACIÓN CON EL CONTENIDO.

Primeras experiencias y acercamiento al tema. Exploración de experiencias personales que se relacionan con el contenido. Pasar de lo desconocido al conocimiento gradual de algo. Es el nivel de asimilación básico (primario) que garantiza el tránsito del estudiante hacia el nivel de asimilación reproductivo. Se realizan lecturas explorativas.

Los conocimientos previos del estudiante son de gran relevancia, para su posterior comprensión del contenido y más aún si es capaz relacionar los conocimientos precedentes con el contenido estudiar. En este sentido la comprensión será asequible ya que el estudiante procesa simultáneamente los datos que provienen del texto y los que proceden de él como lector.

**Primeramente**, se trabaja con un sistema de tareas docentes para el **diagnóstico y aseguramiento** del nivel de partida o aseguramiento de las condiciones previas en la clase, en dependencia del momento en que se encuentra

el programa de estudio se pueden realizar las siguientes tareas: **Verifique** el nivel de conocimientos y habilidades que tienen los estudiantes acerca de:

- a) Los paradigmas de la programación;
- b) Los algoritmos expresados en pseudocódigos y diagramas UML y las estructuras algorítmicas;
- c) Las diferentes etapas en la realización de un programa;
- d) Los conceptos, procedimientos computacionales y habilidades asociadas con el contenido a estudiar, que sirven de base para introducir la POO (Ruiz & Beltrán, 2021).
- e) Los fundamentos de la programación, de modo que los alumnos puedan pasar sin dificultad a otros enfoques, como la programación orientada a objetos (Spyropoulou, et al., 2015).

A continuación, se procede con los sistemas de tareas docentes para el nivel de asimilación familiarización.

#### SISTEMA DE TAREAS DOCENTES

1. **Realización de una lectura exploratoria** del capítulo - Introducción a la Programación Orientada a Objetos, consultando el libro electrónico Pseudocódigos, diagramas UML y códigos en el lenguaje de programación C# para la solución de problemas, u otro que oriente el profesor. Se orienta una segunda lectura procurando una mayor comprensión del texto.
2. **Determinación y toma de notas** en el cuaderno acerca de lo que se ha considerado esencial en las lecturas. Escribe las palabras, frases o inquietudes que necesites preguntar e investigar posteriormente;
3. **Cambio del color** de texto a rojo en los párrafos donde se definen los conceptos “POO”, “Clase”, “Objetos”, “Programación estructurada”, “Atributos o campos”, “Métodos”, “Relaciones entre clases”, “Herencia”, “Diagrama UML”;
4. **Cambio del color** de texto a azul en cada lugar donde se muestre el código para la creación de un objeto en C#;
5. **Identificación** de las partes del texto donde se representan clases con sus componentes en el diagrama UML;
6. **Identificación** en el texto de las relaciones representadas a través de diagramas de clases UML, en particular, las de asociación y herencia;
7. **Reconocimiento** en el texto donde se representan relaciones entre clases “uno a muchos y de muchos a muchos”;

8. **Identificación** en el texto en el que se menciona un paradigma anterior al de la POO. **¿Diga cuál es y en qué consiste?;**
9. **Determinación de 5 objetos** presentes en el aula o laboratorio de estudio.

Las tareas antes expresadas pueden realizarse sustituyendo la lectura exploratoria por la visualización de un video donde se introduzca la POO.

1. **Realización de una visualización exploratoria** del video. Vuelve a visualizarlo procurando una mayor comprensión;
2. **Determinación y toma de notas** en tu cuaderno acerca de lo que has considerado esencial en tus visualizaciones. Escribe las palabras, frases o inquietudes que necesites preguntar e investigar posteriormente;
3. **Detén la corrida del video y anota las ideas cuando** se definan los conceptos "POO", "Clase", "Objetos", "Programación estructurada", "Atributos o campos", "Métodos", "Relaciones entre clases", "Herencia", "Diagrama UML";
4. **Identificación en el video** en el cual se muestre el código para la creación de un objeto en C#;
5. **Identificación** en las partes del video en el cual se representan clases con sus componentes en el diagrama UML;
6. **Identificación** en el video de las relaciones representadas a través de diagramas de clases UML, en particular, las de asociación y herencia;
7. **Reconocimiento** en el video en el que se representan relaciones entre clases "1 a muchos y de muchos a muchos";
8. **Reconocimiento** en el video donde se menciona un paradigma anterior al de la POO. **¿Diga cuál es y en qué consiste?;**
9. **Determinación de 5 objetos** presentes en el aula o laboratorio de estudio.

#### NIVEL DE ASIMILACIÓN: REPRODUCCIÓN.

Se caracteriza por actividades de reproducción (copiar, repetir, transcribir, imitar, producir de nuevo, representar) del contenido en estudio, engloba desde la réplica de un modelo hasta su reproducción a base de memoria, aun cuando este no haya sido interpretado. Su desarrollo garantiza el tránsito exitoso del estudiante al nivel de aplicación.

Antes de la puesta en práctica del sistema de tareas docentes se requiere que el profesor realice las actividades

de diagnóstico y aseguramiento del nivel de partida para este nivel.

#### SISTEMA DE TAREAS DOCENTES

1. **Expresé** con sus palabras y luego compruebe si lo formulado por usted coincide con lo expresado en su bibliografía a orientada:

- a) Programación estructurada;
- b) POO;
- c) Clases;
- d) Objetos;
- e) Métodos (funcionalidades)
- f) Relación de herencia;
- g) Relación de asociación;
- h) Relación de uno a muchos;
- i) Relación de mucho a muchos;

2. **Lea e interprete** un problema (expresado en un texto o un diagrama) y **muestre**:

- a) Las clases que intervienen en su solución;
- b) Los atributos y métodos (funcionalidades) de cada clase;
- c) El código de programación para cada clase.

3. **Examine** un código de programación y responda:

- a) **Expresé** si se corresponde con el paradigma de la POO o la Programación estructurada;
- b) **Verifique** si se corresponde con otro paradigma no expresado anteriormente;
- c) Si es el paradigma de la POO **identifique** la clase con sus atributos, métodos y **representela** en un diagrama de clases UML;
- d) Si es el paradigma de Programación estructurada **identifique** el nombre de sus módulos o funciones.

4. **Expresado** por el profesor la existencia de la clase Persona. **Representela** en un diagrama UML y **escriba** el código de programación que permite crear un objeto;

5. **Conocido** el diagrama UML de la clase Estudiantes, con todos sus elementos esenciales, **escriba** el código de programación que se corresponde con el diagrama;

6. **Expresé** con sus palabras lo que entiendes por relación de asociación y de herencia;

7. **Expresé** con sus palabras lo que entiende por relaciones de clases de uno a uno, uno a muchos y muchos a muchos.

## NIVEL DE ASIMILACIÓN: APLICACIÓN.

Se caracteriza por la aplicación (uso, empleo) de conocimientos y habilidades a la solución de ejercicios, problemas u otras situaciones prácticas. Realización de procedimientos apropiados para alcanzar una meta. Empleo o puesta en práctica de algo. Incluye habilidades intelectuales como la comparación, clasificación y valoración. La explicación de hechos y procesos a partir de teorías (Rojas & Camejo, 2010). Su desarrollo garantiza el tránsito con buenos resultados del estudiante al nivel de creación (García-García, et al., 2019).

Antes de la puesta en práctica del sistema de tareas docentes se requiere que el profesor realice las actividades de diagnóstico y aseguramiento del nivel de partida para este nivel (Rodríguez Hernández & Feliciano García, 2005).

## SISTEMA DE TAREAS DOCENTES

1. **Observe** el siguiente diagrama de clases UML y a continuación:

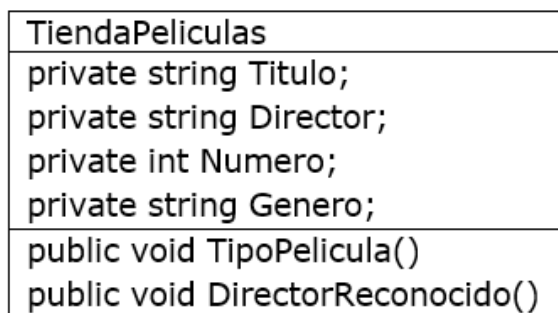


Figura 1. Representación UML de la clase TiendaPelículas.

Fuente: elaboración propia

**Escriba** el código de programación que le corresponde.

2. Se desea elaborar una aplicación para el tratamiento de números enteros que permita determinar, si un número es par o impar, si es positivo o negativo, si es o no primo, así como que permita calcular su cuadrado.

**a) Modele** con el empleo del diagrama de clases UML, la clase que interviene en el problema, así como sus atributos y responsabilidades (Figura 1 y 2)

**b) Escriba** el código de programación que se corresponde con el diagrama UML.

3. **Cree un arreglo** de n - elementos de tipo entero (n se ingresa por teclado). **Muestre** cuantos elementos son superiores a 100. Emplear una estructura repetitiva a fin

de recorrer el arreglo. Solucionar el problema mediante la P.O.O

**a) Escriba el algoritmo** en pseudocódigo que resuelve este problema;

**b) Verifique la validez** del algoritmo con valores de entrada, realice su corrida;

**c) Realice la transición** del algoritmo al código de programación;

**d) Realice las pruebas necesarias** al programa para verificar su validez.

4. **Resuelva** el problema anterior empleando una lista en vez de un arreglo.

**a) Realice las pruebas necesarias** al programa para verificar su validez;

5. Conocido el tiempo de 10 atletas que corren distancias de fondo (10 000 metros). **Realice el algoritmo, verifique la validez** del algoritmo con valores de entrada, realice su corrida y seguidamente el código para:

a) El tiempo promedio de los atletas;

b) La cantidad de días mayor que el promedio;

c) Si está apto o no para la prueba de los 5 Km sabiendo que si alguno de los tiempos es mayor que 16 minutos no lo está;

d) Realice las pruebas necesarias al programa para verificar su validez.

6. **Resuelva** el problema anterior empleando una lista en vez de un arreglo.

7. Para el dominio del presupuesto de una empresa se desea controlar los datos relacionados con cada uno de sus contratos. De cada contrato se conoce: un código, el plan a ingresar (valor monetario), el ingreso actual (contiene el valor monetario actual que se ha ingresado por ese contrato hasta el momento) y el nombre del gestor del contrato. Para el trabajo de la empresa se requiere conocer el porcentaje de cumplimiento del contrato.

**a) Modele** la clase correspondiente a través del diagrama de clases UML (Figura 1 y 2);

**b) Escriba** el código de programación que se corresponde con el diagrama UML.

8. Dado el siguiente código de programación en C#:

```
class Motor
{
    //área de atributos
    private bool enciendeMotor;

    //área de métodos
```

```
public Motor(int cantCombustible)
{
    //validar si existe combustible
    if (cantCombustible>0)
        this.enciendeMotor = true;
    else
        this.enciendeMotor = false;
}
```

- Explique lo que sucede al realizar su corrida;
- Verifique la sintaxis de cada línea de código;
- Represente este código en un diagrama de clases UML;
- Realice el algoritmo que se corresponde con el código del método Motor;
- Verifique la validez del algoritmo con valores de entrada, realice su corrida.

#### NIVEL DE ASIMILACIÓN: CREACIÓN.

Es la creación y gestión del conocimiento. La capacidad de conocer, actuar y transformar la realidad en cada contexto, teniendo como premisas, la autonomía, autorregulación y la cooperación en el aprendizaje (Rumayor, et. al, 2011). Es hacer o rehacer algo con nuevos matices. Es crear objetos, resolver problemas complejos con novedosas inventivas. Establecer estrategias de solución a problemas. Es el nivel de asimilación superior (Rodríguez Hernández & Feliciano García, 2005; Gargallo-López et al., 2020).

Antes de la puesta en práctica del sistema de tareas docentes se requiere que el profesor realice las actividades de diagnóstico y aseguramiento del nivel de partida para este nivel.

#### SISTEMA DE TAREAS DOCENTES

- Dado el siguiente código de programación en C#:

```
class Motor
{
    //área de atributos
    private bool enciendeMotor;
    //área de métodos
    public Motor(int cantCombustible)
    {
        //validar si existe combustible
        if (cantCombustible>0)
            this.enciendeMotor = true;
        else
            this.enciendeMotor = false;
    }
}
```

- Verifique la sintaxis en cada línea de código;
- Reescribalo de nuevo con una lógica de programación diferente y con menos números de líneas de código, si fuese posible.
- Realice las pruebas necesarias al programa para verificar su validez.

- Comenzando con el siguiente diagrama UML de clases, **crea**:

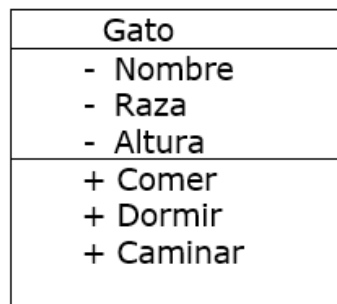


Figura 2. Representación UML de la clase Gato.

Fuente: elaboración propia

En un diagrama de clases UML:

- La clase Perro;
- La superclase Animal (clase madre);
- La relación de herencia entre las tres clases;
- El código de la relación de herencia representada, en el lenguaje de programación C# u otro si lo prefieres.

- Dada la clase Persona:

```
class Persona
{
    // Campos de la clase Persona
    public string Nombre;
    public int Edad;
    public string CI;

    void Cumpleaños()
    {
        Edad++;
    }
    // Constructor de la clase Persona
    public Persona(string nombre, int edad, string ci)
    {
        Nombre = nombre;
        Edad = edad;
        CI = ci;
    }
}
```



- a) De manera análoga,  **Cree** el código de la clase Trabajador;
- b) Represente** en diagramas de clases UML, las clases Trabajador y Persona (Figura 1 y 2);
- c) Represente** la relación de herencia entre ambas clases;
- d) Verifique la sintaxis** de cada línea de código.
4.  **Cree** los diagramas de clases UML para representar, independientemente, las clases Vehículo; Auto y Camioneta, con los atributos seleccionados por usted.  **Establezca** la posible relación entre las clases y a continuación:
- a) Realice** el diagrama de clases UML que se corresponde con la relación entre las clases antes modeladas;
- b) Realice** el código de programación en C# que se corresponde con la relación identificada por usted.
5.  **Cree** los diagramas de clases UML para representar, independientemente, las clases Hospital; Médico y Ambulancia con los atributos escogidos por usted.  **Establezca** la posible relación entre las clases y a continuación:
- a) Cree** el diagrama de clases UML que representa la relación de las clases antes modeladas;
- b) Cree** el código de programación en C# que se corresponde con la relación identificada por usted;
- c) Dibuje** en un papel la interface gráfica que considere más apropiada para este ejemplo.
6. Dado el siguiente código de programación:

```
//Variables Auxiliares
double area = 0, lado = 0;
Console.WriteLine("Determinar el área de un Cuadrado.");

//Solicitando datos
Console.Write("Ingrese el lado del cuadrado: ");

//Convierte de texto a tipo entero
lado = Int32.Parse(Console.ReadLine());

//Realiza la operación
area = lado * lado;

//Imprimiendo el resultado
Console.WriteLine("El área del Cuadrado: " + area);

//El programa espera a que se reciba una tecla
Console.ReadKey();
```

- a) Verifique** su sintaxis;
- b) Redacte** el problema que se corresponde con esa solución;
- c) Realice** la transición del código de programación a un algoritmo expresado en pseudocódigo;
- d) Reescriba** el programa aplicando lo aprendido en POO, es decir, creando una clase y dentro de ella un método;
- e) Represente** la clase en un diagrama UML.

Finalmente, posterior al nivel de creación, el profesor puede orientar sistemas de tareas docentes de carácter investigativas, entre ellas:

- a) Investigue** qué es el polimorfismo en POO y qué ventajas proporciona en la elaboración de un programa;
- b) Indague** a cerca de las diferencias entre las relaciones de agregación y composición entre clases;
- c) Examine** la relación de herencia y compárela con sus conocimientos cotidianos;
- d) Identifique** qué lenguajes de programación admiten la herencia múltiple y diga en qué consiste.

## RESULTADOS Y DISCUSIÓN.

Se ha constatado, en controles a clases que más del 90% de los profesores que imparten POO carecen de la suficiente experiencia para estructurar, apropiadamente, sistemas de tareas docentes y menos aún, relacionarlas con los niveles de asimilación del contenido. Cuestión que está en correspondencia con que los docentes (más del 90%) desconocen la existencia de niveles de asimilación y en particular el nivel de familiarización con el contenido, por lo que pasan directamente a la solución de tareas reproductivas y de aplicación del contenido sin transitar por las anteriores.

Lo anteriormente expuesto se corresponde con los resultados académicos históricos en las universidades José Eduardo Dos Santos de Angola y la Universidad Tecnológica de la Habana “José Antonio Echeverría”, en materia de POO, inferiores al 60% en los primeros exámenes parciales y en la medida que avanza el curso muestran una ligera mejoría.

Pese a que las indagaciones empíricas realizadas no aportan datos suficientes acerca de cómo se realiza este proceso en otras universidades, la introducción de grupos de tareas en diferentes escenarios en las Facultades de Ingeniería Informática de las referidas universidades, en las cuales el autor participa directamente, muestra evidencias de efectividad en la formación informática del ingeniero.

Se ha observado que la aplicación de grupos de tareas docentes, aun con las insuficiencias detectadas, propician avances en la adquisición de conocimientos, desarrollo de habilidades y formación de valores en los estudiantes.

En este sentido, los análisis docentes de los resultados mostraron una tendencia a la recuperación, tanto desde lo cuantitativo como en lo cualitativo, evidenciados en:

- a) El autor, como participante activo en las evaluaciones parciales y finales ha constatado el aumento de la calidad de las respuestas al tribunal y el software elaborado, en la defensa de los proyectos informáticos, distribuidos en equipos de cuatro integrantes. En este punto, 65 obtuvieron calificaciones de excelente en el curso escolar 2017-2018 para un 72% y 69 en 2018-2019 para un 76%;
- b) El mejoramiento de los informes digitales, tanto en sus estructuras como en la calidad de sus contenidos expuestos. En este aspecto 82 de 90 recibieron notas sobresalientes para un 91% en el curso 2017-2018;
- c) El ascenso de los resultados cuantitativos de la disciplina, en lo que concierne a las calificaciones finales y disminución de los reprobados. Más del 75% de los

estudiantes demostraron haber cumplido los objetivos del programa en primera convocatoria de exámenes en los cursos escolares antes referido;

d) Avances en la capacidad transferir pseudocódigos a códigos de programación. Un total de 68 estudiantes demostraron haber desarrollado esta habilidad para un 75%;

e) La aprobación de los docentes, en reuniones metodológicas realizadas los cursos 2017 – 2018 y 2018 – 2019, donde se analizaron los elementos esenciales de los sistemas de tareas elaborados. En este punto, el 100% de los profesores consideraron excelente la propuesta;

f) No obstante, en más de 30 controles realizados, aunque se apreciaron clases con una mejor estructuración metodológica y una tendencia a optimizar el método de estudio para la resolución de problemas, se observaron insuficiencias, en un 58% de los estudiantes en la realización del estudio independiente y el desarrollo de interfaces gráficas.

El autor después de exponer los sistemas de tareas docentes y valorar los resultados parciales obtenidos, infiere que su aplicación influye positivamente en el perfeccionamiento del PEA de la POO, en las carreras de Ingeniería Informática e Industrial y como consecuencia al logro de una formación informática superior en ambas instituciones universitarias. Además, los sistemas de tareas docentes constituyen una contribución al enriquecimiento de la didáctica del PEA de la POO en las carreras de ingenierías.

### Resultados esperados con la aplicación sistemática de los Sistemas de Tareas Docentes

La aplicación continua, creativa y contextualizada de los sistemas de tareas docentes presentados en este artículo debe mejorar considerablemente el aprendizaje de los estudiantes en la POO. Se espera:

- a) Una atinada interpretación del texto o problema a resolver y el establecimiento de planes de acciones antes de entrar en la fase de codificación;
- b) El aumento gradual del interés por elaborar el algoritmo que conduce a la solución del problema y su posterior validación antes de codificarlo a un lenguaje seleccionado;
- c) La conversión de los diagramas de flujo a pseudocódigo y viceversa;
- d) El dominio de otros paradigmas y su transición al de la POO;

- e) El cuidado y la debida atención a las exigencias de sintaxis del lenguaje de programación, lo cual puede producir frustraciones en el estudiante;
- f) La atención a las diferencias individuales de los estudiantes;
- g) La optimización de los códigos de programación y la realización de las pruebas necesarias al programa para verificar su validez;
- h) El mejoramiento cuantitativo y cualitativo en las evaluaciones parciales y finales;
- i) Una comprensión superior de los conceptos fundamentales de la POO, antes de codificar, (Jang, Lee, & Kim, 2015)
- j) Disminuir el grado de deserción universitaria en estudiantes que reciben estudios a profundidad de POO.
- k) El avance del estudiante en una gradual formación de la responsabilidad y autonomía al gestionar sus aprendizajes significativos, siempre que se cuente con las ayudas necesarias.

#### Temas para continuar indagando

1. La relación que debe existir en el proceso de enseñanza y aprendizaje de la POO y las funciones didácticas:
  - a) Aseguramiento del nivel de partida;
  - b) Motivación;
  - c) Tratamiento de la nueva materia;
  - d) Consolidación;
  - e) Control.
2. Si la introducción gradual de los conceptos de la POO en la enseñanza secundaria y preuniversitaria contribuye a la obtención de mejores resultados académicos en las universidades.

#### CONCLUSIONES

La introducción en la práctica educativa de grupos de tareas docentes, contribuye a la adquisición de conocimientos, desarrollo de habilidades y formación de valores en los estudiantes (Arredondo - Tapia & García-Ponce, 2021).

La aplicación creativa y contextualizada de los sistemas de tareas docentes presentados en este artículo perfeccionará el aprendizaje de la POO y como consecuencia el rendimiento cuantitativo y cualitativo en las evaluaciones parciales y finales (Rodríguez Hernández & Feliciano García, 2005).

La realización de tareas docentes, obviando los niveles de asimilación, transgrede las etapas del proceso de

enseñanza y aprendizaje de la POO, y como consecuencia, se obtendrán resultados docentes inesperados e indeseados, en cambio, su elaboración y empleo, conforme a los niveles de aprovechamiento del contenido y con una adecuada relación teórica – práctica, propiciará un aprendizaje desarrollador y altos resultados académicos en los estudiantes.

Un oportuno diagnóstico y aseguramiento del nivel de partida en cada clase, es necesario antes de emprender un sistema de tareas docentes, no solo en el nivel de familiarización del contenido, sino en los demás niveles.

Una vez alcanzado el nivel de asimilación “creación”, se recomienda introducir sistemas de tareas encaminadas a la investigación, las cuales contribuirán al desarrollo de la autogestión del conocimiento en estudiantes y profesores y al logro de una formación informática superior.

#### REFERENCIAS BIBLIOGRÁFICAS

- Arredondo-Tapia, D. I., & García-Ponce, E. E. (2021). Role of Task Repetition and Content Familiarity in EFL Students' Fluency and Accuracy in Narrative Tasks: A Case Study. *Journal of Language and Education*, 7(2), 45-63. <https://doi.org/10.17323/jle.2021.11202>
- Díaz-Leiva, T., & Chamorro-Atalaya, O. (2020). Analysis of Learning Difficulties in Object Oriented Programming in Systems Engineering Students at UNTELS. *Advances in Science, Technology and Engineering Systems Journal*. 5( 6), 1704-1709.
- Fonden, J. C. (2006). *Una estrategia didáctica interdisciplinaria para el proceso de enseñanza - aprendizaje de la computación de los bachilleres técnicos en la especialidad de informática*. (Tesis Doctoral). Universidad de Ciencias Pedagógicas Enrique José Varona).
- Fonden, J. C., Stuart M. L., & Rodríguez, L. (2018). La algoritmización: requisito necesario para la solución de problemas con el empleo de un lenguaje de programación. *Luz*. 17(3), 30-43.
- Fuentes, J.I., & Moo, M. (2017). Dificultades de aprender a programar. *Educación en Ingeniería*, 12(24), 76-82.
- García-Valcárcel, A., Muñoz-Repiso, A., & Basilotta-Gómez, P. (2017). Aprendizaje Basado en Proyectos (ABP): evaluación desde la perspectiva de alumnos de Educación Primaria. *Revista de Investigación Educativa*, 35(1), 113-131.

- García-García, F. J., Quesada-Armenteros, A., Romero-Ariza, M., & Abril-Gallego, A. M. (2019). Promoting inquiry in Mathematics and Science: professional development of Primary and Secondary school teachers. *Educación XX1*, 22(2), 335-359.
- Gargallo-López, B., Pérez-Pérez, C., García-García, F.J., Giménez-Beut, J.A., & Portillo-Poblador, N. (2020). The skill of learning to learn at university. Proposal for a theoretical model. *Educación XX1*, 23(1), 19-44.
- Jang, Y. J., Lee, W. G., & Kim, J. M. (2015). Assessing the usefulness of object-based programming education using arduino. *Indian Journal of Science and Technology*, 8, 89-96.
- Martínez, L. A., Iglesias, M., & Espinosa, A. (2010). Sistema de tareas docentes para la formación de la habilidad toma de decisiones médicas mediante el método clínico. *Medisur*. 8(6). 426-436.
- Martínez, S., Massip, A., & Pérez, F. J. (2021). El estudio y trabajo independientes en la mira de la educación médica superior cubana. *Educación Médica Superior*. 35(1), 1-20.
- Martínez, L., E., Echeverría L., & Hernández, L. (2021). La tarea docente. Tratamiento de los contenidos biológicos en secundaria básica. *Mendive. Revista de Educación*, 19(1), 30-40.
- Pineda-Alfonso, J. A., & Duarte-Piña, O. M. (2020). Pedagogical conceptions of university teaching: a starting point for teacher change. *Educación XX1*, 23(2), 95-118.
- Rodríguez Hernández, J. A., & Feliciano García, L. (2005). Toma de decisiones y tareas del profesorado de Educación Primaria. *Revista de Investigación Educativa*, 23(2), 483-499.
- Rodríguez, J.M. (2009). Competencias educativas vinculadas a la profesión docente. *Revista Educación y Pedagogía*, 21(53), 113-123.
- Ruiz, R., & Beltrán, C. (2021). Las funciones didácticas en la enseñanza de la Matemática. *EduSol*. 21(75), 1-14.
- Rojas, D. A., & Camejo, M. (2010). Niveles de asimilación y niveles de desempeño cognitivo: reflexionemos. *Dialnet*. 8(1), 1-6.
- Rumayor, L., de las Heras, A. M., & Muñoz, Y. (2011). La creación y gestión del conocimiento en la enseñanza superior: la autonomía, autorregulación y cooperación en el aprendizaje. *Universia*. 4(2), 105-120.
- Schnitzler, N., & Núñez, P. (2021). Hacia una evaluación de la reflexión pedagógica desde la escritura académica. *Magis. Revista Internacional de Investigación en Educación*, 14, 1-25.
- Spyropoulou, N., Demopoulou, G., Pierrakeasa, C., Koutsonikosa, I., & Kameas, A. (2015). Developing a Computer Programming MOOC. *Procedia Computer Science* 65, 182 – 191.