

I. Guía pedagógica del módulo Programación básica orientada a objetos

Contenido

	Pág.
I. Guía pedagógica	
1. Descripción	3
2. Datos de identificación de la norma	4
3. Generalidades pedagógicas	5
4. Enfoque del módulo	12
5. Orientaciones didácticas y estrategias de aprendizaje por unidad	13
6. Prácticas/ejercicios/problemas/actividades	23
II. Guía de evaluación	42
7. Descripción	43
8. Tabla de ponderación	47
9. Materiales para el desarrollo de actividades de evaluación	48
10. Matriz de valoración o rúbrica	49

1. Descripción

La Guía Pedagógica es un documento que integra elementos técnico-metodológicos planteados de acuerdo con los principios y lineamientos del **Modelo Académico del Conalep** para orientar la práctica educativa del docente en el desarrollo de competencias previstas en los programas de estudio.

La finalidad que tiene esta guía es facilitar el aprendizaje de los alumnos, encauzar sus acciones y reflexiones y proporcionar situaciones en las que desarrollará las competencias. El docente debe asumir conscientemente un rol que facilite el proceso de aprendizaje, proponiendo y cuidando un encuadre que favorezca un ambiente seguro en el que los alumnos puedan aprender, tomar riesgos, equivocarse extrayendo de sus errores lecciones significativas, apoyarse mutuamente, establecer relaciones positivas y de confianza, crear relaciones significativas con adultos a quienes respetan no por su estatus como tal, sino como personas cuyo ejemplo, cercanía y apoyo emocional es valioso.

Es necesario destacar que el desarrollo de la competencia se concreta en el aula, ya que **formar con un enfoque en competencias significa crear experiencias de aprendizaje para que los alumnos adquieran la capacidad de movilizar, de forma integral, recursos que se consideran indispensables para saber resolver problemas en diversas situaciones o contextos**, e involucran las dimensiones cognitiva, afectiva y psicomotora; por ello, los programas de estudio, describen las competencias a desarrollar, entendiéndolas como la combinación integrada de conocimientos, habilidades, actitudes y valores que permiten el logro de un desempeño eficiente, autónomo, flexible y responsable del individuo en situaciones específicas y en un contexto dado. En consecuencia, la competencia implica la comprensión y transferencia de los conocimientos a situaciones de la vida real; ello exige relacionar, integrar, interpretar, inventar, aplicar y transferir los saberes a la resolución de problemas. Esto significa que **el contenido, los medios de enseñanza, las estrategias de aprendizaje, las formas de organización de la clase y la evaluación se estructuran en función de la competencia a formar**; es decir, el énfasis en la proyección curricular está en lo que los alumnos tienen que aprender, en las formas en cómo lo hacen y en su aplicación a situaciones de la vida cotidiana y profesional.

Considerando que el alumno está en el centro del proceso formativo, se busca acercarle elementos de apoyo que le muestren qué **competencias** va a desarrollar, cómo hacerlo y la forma en que se le evaluará. Es decir, mediante la guía pedagógica el alumno podrá **autogestionar su aprendizaje** a través del uso de estrategias flexibles y apropiadas que se transfieran y adopten a nuevas situaciones y contextos e ir dando seguimiento a sus avances a través de una autoevaluación constante, como base para mejorar en el logro y desarrollo de las competencias indispensables para un crecimiento académico y personal.

2. Datos de Identificación de la Norma

Título:			
Unidad (es) de Norma Técnica de Competencia Laboral:			
Código:		Nivel de competencia:	

3. Generalidades pedagógicas

Con el propósito de difundir los criterios a considerar en la instrumentación de la presente guía entre los docentes y personal académico de planteles y Colegios Estatales, se describen **algunas consideraciones** respecto al desarrollo e intención de las competencias expresadas en los módulos correspondientes a la formación básica, propedéutica y profesional.

Los principios asociados a la **concepción constructivista del aprendizaje** mantienen una estrecha relación con los de la **educación basada en competencias**, la cual se ha concebido en el Colegio como el enfoque idóneo para orientar la formación ocupacional de los futuros profesionales técnicos y profesionales técnicos-bachilleres. Este enfoque constituye una de las opciones más viables para lograr la vinculación entre la educación y el sector productivo de bienes y servicios.

En los programas de estudio se proponen una serie de contenidos que se considera conveniente abordar para obtener los **Resultados de Aprendizaje establecidos**; sin embargo, se busca que este planteamiento le dé al docente la posibilidad de **desarrollarlos con mayor libertad y creatividad**.

En este sentido, se debe considerar que el papel que juegan el alumno y el docente en el marco del Modelo Académico del Conalep tenga, entre otras, las siguientes características:

El alumno:	El docente:
<ul style="list-style-type: none"> ❖ Mejora su capacidad para resolver problemas. ❖ Aprende a trabajar en grupo y comunica sus ideas. ❖ Aprende a buscar información y a procesarla. ❖ Construye su conocimiento. ❖ Adopta una posición crítica y autónoma. ❖ Realiza los procesos de autoevaluación y coevaluación. 	<ul style="list-style-type: none"> ❖ Organiza su formación continua a lo largo de su trayectoria profesional. ❖ Domina y estructura los saberes para facilitar experiencias de aprendizaje significativo. ❖ Planifica los procesos de enseñanza y de aprendizaje atendiendo al enfoque por competencias, y los ubica en contextos disciplinares, curriculares y sociales amplios. ❖ Lleva a la práctica procesos de enseñanza y de aprendizaje de manera efectiva, creativa e innovadora a su contexto institucional. ❖ Evalúa los procesos de enseñanza y de aprendizaje con un enfoque formativo. ❖ Construye ambientes para el aprendizaje autónomo y colaborativo. ❖ Contribuye a la generación de un ambiente que facilite el desarrollo sano e integral de los estudiantes. ❖ Participa en los proyectos de mejora continua de su escuela y apoya la gestión institucional.

En esta etapa se requiere una mejor y mayor organización académica que apoye en forma relativa la actividad del alumno, que en este caso es mucho mayor que la del docente; lo que no quiere decir que su labor sea menos importante. **El docente en lugar de transmitir vertical y unidireccionalmente los conocimientos, es un mediador del aprendizaje**, ya que:

- Planea y diseña experiencias y actividades necesarias para la adquisición de las competencias previstas. Asimismo, define los ambientes de aprendizaje, espacios y recursos adecuados para su logro.
- Proporciona oportunidades de aprendizaje a los estudiantes apoyándose en metodologías y estrategias didácticas pertinentes a los Resultados de Aprendizaje.
- Ayuda también al alumno a asumir un rol más comprometido con su propio proceso, invitándole a tomar decisiones.
- Facilita el aprender a pensar, fomentando un nivel más profundo de conocimiento.
- Ayuda en la creación y desarrollo de grupos colaborativos entre los alumnos.
- Guía permanentemente a los alumnos.
- Motiva al alumno a poner en práctica sus ideas, animándole en sus exploraciones y proyectos.

Considerando la importancia de que el docente planee y despliegue con libertad su experiencia y creatividad para el desarrollo de las competencias consideradas en los programas de estudio y especificadas en los Resultados de Aprendizaje, en las competencias de las Unidades de Aprendizaje, así como en la competencia del módulo; **podrá proponer y utilizar todas las estrategias didácticas que considere necesarias** para el logro de estos fines educativos, con la recomendación de que fomente, preferentemente, las estrategias y técnicas didácticas que se describen en este apartado.

Al respecto, entenderemos como estrategias didácticas los planes y actividades orientados a un desempeño exitoso de los resultados de aprendizaje, que incluyen estrategias de enseñanza, estrategias de aprendizaje, métodos y técnicas didácticas, así como, acciones paralelas o alternativas que el docente y los alumnos realizarán para obtener y verificar el logro de la competencia; bajo este tenor, **la autoevaluación debe ser considerada también como una estrategia por excelencia para educar al alumno en la responsabilidad y para que aprenda a valorar, criticar y reflexionar sobre el proceso de enseñanza y su aprendizaje individual.**

Es así como la selección de estas estrategias debe orientarse hacia un enfoque constructivista del conocimiento y estar dirigidas a que **los alumnos observen y estudien su entorno**, con el fin de generar nuevos conocimientos en contextos reales y el desarrollo de las capacidades reflexivas y críticas de los alumnos.

Desde esta perspectiva, a continuación se describen brevemente los tipos de aprendizaje que guiarán el diseño de las estrategias y las técnicas que deberán emplearse para el desarrollo de las mismas:

TIPOS DE APRENDIZAJES

Aprendizaje Significativo

Se fundamenta en una concepción constructivista del aprendizaje, la cual se nutre de diversas concepciones asociadas al cognoscitivismo, como la teoría psicogenética de Jean Piaget, el enfoque sociocultural de Vygotsky y la teoría del aprendizaje significativo de Ausubel.

Dicha concepción sostiene que el ser humano tiene la disposición de **aprender verdaderamente sólo aquello a lo que le encuentra sentido** en virtud de que está vinculado con su entorno o con sus conocimientos previos. Con respecto al comportamiento del alumno, se espera que sean capaces de desarrollar aprendizajes significativos, en una amplia gama de situaciones y circunstancias, lo cual equivale a “**aprender a aprender**”, ya que de ello depende la construcción del conocimiento.

Aprendizaje Colaborativo.

El aprendizaje colaborativo puede definirse como el conjunto de métodos de instrucción o entrenamiento para uso en grupos, así como de estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social). En el aprendizaje colaborativo **cada miembro del grupo es responsable de su propio aprendizaje, así como del de los restantes miembros del grupo** (Johnson, 1993.)

Más que una técnica, el aprendizaje colaborativo es considerado una filosofía de interacción y una forma personal de trabajo, que implica el manejo de aspectos tales como el **respeto a las contribuciones y capacidades individuales de los miembros del grupo** (Maldonado Pérez, 2007). Lo que lo distingue de otro tipo de situaciones grupales, es el desarrollo de la interdependencia positiva entre los alumnos, es decir, de una toma de conciencia de que **sólo es posible lograr las metas individuales de aprendizaje si los demás compañeros del grupo también logran las suyas**.

El aprendizaje colaborativo surge a través de transacciones entre los alumnos, o entre el docente y los alumnos, en un proceso en el cual cambia la responsabilidad del aprendizaje, del docente como experto, al alumno, y asume que el docente es también un sujeto que aprende. Lo más importante en la formación de grupos de trabajo colaborativo es vigilar que los elementos básicos estén claramente estructurados en cada sesión de trabajo. Sólo de esta manera se puede lograr que se produzca, tanto el esfuerzo colaborativo en el grupo, como una estrecha relación entre la colaboración y los resultados (Jonson & F. Jonson, 1997).

Los elementos básicos que deben estar presentes en los grupos de trabajo colaborativo para que éste sea efectivo son:

- la interdependencia positiva.
- la responsabilidad individual.
- la interacción promotora.
- el uso apropiado de destrezas sociales.
- el procesamiento del grupo.

Asimismo, el trabajo colaborativo se caracteriza principalmente por lo siguiente:

- Se desarrolla mediante acciones de cooperación, responsabilidad, respeto y comunicación, en forma sistemática, entre los integrantes del grupo y subgrupos.
- Va más allá que sólo el simple trabajo en equipo por parte de los alumnos. Básicamente se puede orientar a que los alumnos intercambien información y trabajen en tareas hasta que todos sus miembros las han entendido y terminado, aprendiendo a través de la colaboración.
- Se distingue por el desarrollo de una interdependencia positiva entre los alumnos, en donde se tome conciencia de que sólo es posible lograr las metas individuales de aprendizaje si los demás compañeros del grupo también logran las suyas.
- Aunque en esencia esta estrategia promueve la actividad en pequeños grupos de trabajo, se debe cuidar en el planteamiento de las actividades que cada integrante obtenga una evidencia personal para poder integrarla a su portafolio de evidencias.

Aprendizaje Basado en Problemas.

Consiste en la presentación de **situaciones reales o simuladas** que requieren la aplicación del conocimiento, en las cuales el **alumno debe analizar la situación y elegir o construir una o varias alternativas para su solución** (Díaz Barriga Arceo, 2003). Es importante aplicar esta estrategia ya que **las competencias se adquieren en el proceso de solución de problemas** y en este sentido, el alumno aprende a solucionarlos cuando se enfrenta a problemas de su vida cotidiana, a problemas vinculados con sus vivencias dentro del Colegio o con la profesión. Asimismo, el alumno se apropia de los conocimientos, habilidades y normas de comportamiento que le permiten la aplicación creativa a nuevas situaciones sociales, profesionales o de aprendizaje, por lo que:

- Se puede trabajar en forma individual o de grupos pequeños de alumnos que se reúnen a analizar y a resolver un problema seleccionado o diseñado especialmente para el logro de ciertos resultados de aprendizaje.
- Se debe presentar primero el problema, se identifican las necesidades de aprendizaje, se busca la información necesaria y finalmente se regresa al problema con una solución o se identifican problemas nuevos y se repite el ciclo.
- Los problemas deben estar diseñados para motivar la búsqueda independiente de la información a través de todos los medios disponibles para el alumno y además generar discusión o controversia en el grupo.
- El mismo diseño del problema debe estimular que los alumnos utilicen los aprendizajes previamente adquiridos.
- El diseño del problema debe comprometer el interés de los alumnos para examinar de manera profunda los conceptos y objetivos que se quieren aprender.
- El problema debe estar en relación con los objetivos del programa de estudio y con problemas o situaciones de la vida diaria para que los alumnos encuentren mayor sentido en el trabajo que realizan.
- Los problemas deben llevar a los alumnos a tomar decisiones o hacer juicios basados en hechos, información lógica y fundamentada, y obligarlos a justificar sus decisiones y razonamientos.
- Se debe centrar en el alumno y no en el docente.

TÉCNICAS

Método de proyectos.

Es una técnica didáctica que incluye actividades que pueden requerir que los alumnos investiguen, construyan y analicen información que coincida con los objetivos específicos de una tarea determinada en la que se organizan actividades desde una perspectiva experiencial, donde el alumno aprende a través de la práctica personal, activa y directa con el propósito de aclarar, reforzar y construir aprendizajes (Intel Educación).

Para definir proyectos efectivos se debe considerar principalmente que:

- Los alumnos son el centro del proceso de aprendizaje.
- Los proyectos se enfocan en resultados de aprendizaje acordes con los programas de estudio.
- Las preguntas orientadoras conducen la ejecución de los proyectos.
- Los proyectos involucran múltiples tipos de evaluaciones continuas.
- El proyecto tiene conexiones con el mundo real.
- Los alumnos demuestran conocimiento a través de un producto o desempeño.
- La tecnología apoya y mejora el aprendizaje de los alumnos.
- Las destrezas de pensamiento son integrales al proyecto.

Para el presente módulo se hacen las siguientes recomendaciones:

- Integrar varios módulos mediante el método de proyectos, lo cual es ideal para desarrollar un trabajo colaborativo.
- En el planteamiento del proyecto, cuidar los siguientes aspectos:
 - ✓ Establecer el alcance y la complejidad.
 - ✓ Determinar las metas.
 - ✓ Definir la duración.
 - ✓ Determinar los recursos y apoyos.
 - ✓ Establecer preguntas guía. Las preguntas guía conducen a los alumnos hacia el logro de los objetivos del proyecto. La cantidad de preguntas guía es proporcional a la complejidad del proyecto.
 - ✓ Calendarizar y organizar las actividades y productos preeliminares y definitivos necesarias para dar cumplimiento al proyecto.
- Las actividades deben ayudar a responsabilizar a los alumnos de su propio aprendizaje y a aplicar competencias adquiridas en el salón de clase en proyectos reales, cuyo planteamiento se basa en un problema real e involucra distintas áreas.

- El proyecto debe implicar que los alumnos participen en un proceso de investigación, en el que utilicen diferentes estrategias de estudio; puedan participar en el proceso de planificación del propio aprendizaje y les ayude a ser flexibles, reconocer al "otro" y comprender su propio entorno personal y cultural. Así entonces se debe favorecer el desarrollo de estrategias de indagación, interpretación y presentación del proceso seguido.
- De acuerdo a algunos teóricos, mediante el método de proyectos los alumnos buscan soluciones a problemas no convencionales, cuando llevan a la práctica el hacer y depurar preguntas, debatir ideas, hacer predicciones, diseñar planes y/o experimentos, recolectar y analizar datos, establecer conclusiones, comunicar sus ideas y descubrimientos a otros, hacer nuevas preguntas, crear artefactos o propuestas muy concretas de orden social, científico, ambiental, etc.
- En la gran mayoría de los casos los proyectos se llevan a cabo fuera del salón de clase y, dependiendo de la orientación del proyecto, en muchos de los casos pueden interactuar con sus comunidades o permitirle un contacto directo con las fuentes de información necesarias para el planteamiento de su trabajo. Estas experiencias en las que se ven involucrados hacen que aprendan a manejar y usar los recursos de los que disponen como el tiempo y los materiales.
- Como medio de evaluación se recomienda que todos los proyectos tengan una o más presentaciones del avance para evaluar resultados relacionados con el proyecto.
- Para conocer acerca del progreso de un proyecto se puede:
 - ✓ Pedir reportes del progreso.
 - ✓ Presentaciones de avance,
 - ✓ Monitorear el trabajo individual o en grupos.
 - ✓ Solicitar una bitácora en relación con cada proyecto.
 - ✓ Calendarizar sesiones semanales de reflexión sobre avances en función de la revisión del plan de proyecto.

Estudio de casos.

El estudio de casos es una técnica de enseñanza en la que los alumnos **aprenden sobre la base de experiencias y situaciones de la vida real**, y se permiten así, construir su propio aprendizaje en un contexto que los aproxima a su entorno. Esta técnica se basa en la participación activa y en procesos colaborativos y democráticos de discusión de la situación reflejada en el caso, por lo que:

- Se deben representar situaciones problemáticas diversas de la vida para que se estudien y analicen.
- Se pretende que los alumnos generen soluciones validas para los posibles problemas de carácter complejo que se presenten en la realidad futura.
- Se deben proponer datos concretos para reflexionar, analizar y discutir en grupo y encontrar posibles alternativas para la solución del problema planteado. Guiar al alumno en la generación de alternativas de solución, le permite desarrollar la habilidad creativa, la capacidad de innovación y representa un recurso para conectar la teoría a la práctica real.

- Debe permitir reflexionar y contrastar las propias conclusiones con las de otros, aceptarlas y expresar sugerencias.

El estudio de casos es pertinente usarlo cuando se pretende:

- Analizar un problema.
- Determinar un método de análisis.
- Adquirir agilidad en determinar alternativas o cursos de acción.
- Tomar decisiones.

Algunos teóricos plantean las siguientes fases para el estudio de un caso:

- **Fase preliminar:** Presentación del caso a los participantes
- **Fase de eclosión:** "Explosión" de opiniones, impresiones, juicios, posibles alternativas, etc., por parte de los participantes.
- **Fase de análisis:** En esta fase es preciso llegar hasta la determinación de aquellos hechos que son significativos. Se concluye esta fase cuando se ha conseguido una síntesis aceptada por todos los miembros del grupo.
- **Fase de conceptualización:** Es la formulación de conceptos o de principios concretos de acción, aplicables en el caso actual y que permiten ser utilizados o transferidos en una situación parecida.

Interrogación.

Consiste en llevar a los alumnos a la **discusión y al análisis de situaciones o información**, con base en preguntas planteadas y formuladas por el docente o por los mismos alumnos, con el fin de explorar las capacidades del pensamiento al activar sus procesos cognitivos; se recomienda **integrar esta técnica de manera sistemática y continua** a las anteriormente descritas y al abordar cualquier tema del programa de estudio.

Participativo-vivenciales.

Son un conjunto de elementos didácticos, sobre todo los que exigen un grado considerable de **involucramiento y participación de todos los miembros del grupo** y que sólo tienen como límite el grado de imaginación y creatividad del facilitador.

Los ejercicios vivenciales son una alternativa para llevar a cabo el proceso enseñanza-aprendizaje, no sólo porque facilitan la transmisión de conocimientos, sino porque además permiten **identificar y fomentar aspectos de liderazgo, motivación, interacción y comunicación del grupo**, etc., los cuales son de vital importancia para la organización, desarrollo y control de un grupo de aprendizaje.

Los ejercicios vivenciales resultan ser una situación planeada y estructurada de tal manera que representan una experiencia muy atractiva, divertida y hasta emocionante. El juego significa apartarse, salirse de lo rutinario y monótono, para asumir un papel o personaje a través del cual el individuo pueda manifestar lo que verdaderamente es o quisiera ser sin temor a la crítica, al rechazo o al ridículo.

El desarrollo de estas experiencias se encuentra determinado por los conocimientos, habilidades y actitudes que el grupo requiera revisar o analizar y por sus propias vivencias y necesidades personales.

4. Enfoque del módulo

La competencia que se adquiere con el desarrollo del módulo, implica realizar la programación orientada a objetos, utilizando las estructuras básicas para solucionar problemas de la vida cotidiana.

Las competencias que se pretenden fomentar consideran actividades tales como aplicar las ventajas de manejar objetos y clases, así como utilizar las facilidades del encapsulamiento y herencia en la programación, manejar estas estructuras y reglas, así como utilizar la ventaja de la modularidad en cualquier lenguaje de programación, con la finalidad de incrementar la productividad y confiabilidad en el desarrollo de programas básicos que permitan resolver las tareas de su entorno.

El módulo considera el desarrollo de un proceso formativo secuencial, aprovechando los conocimientos previos del alumno, que le permita realizar actividades profesionales especializadas en pequeñas y medianas empresas, como microempresario o contratista de otras con mayor presencia en el mercado, dedicadas al desarrollo de software, Programación orientada a objetos, desarrollo de sistemas informáticos, instalación y mantenimiento de sistemas. Dicho proceso implica el desarrollo de actividades tales como identificar los fundamentos de la programación orientada a objetos, considerando la estructura de programas bajo este ambiente, manejar los principios de la programación orientada a objetos, manejar los modelos de la programación orientada a objetos en el análisis, diseño y desarrollo de programas, desarrollar programas de aplicación final, empleando los fundamentos y características de un lenguaje Orientado a Objetos para dar solución a problemas de aplicación real así como manejar las estructuras y metodología en cualquier lenguaje de programación.

Dado la naturaleza de formación integral, el módulo también fomenta en el alumno el desarrollo de las competencias disciplinares básicas y genéricas tales como la interpretación y emisión de mensajes pertinentes en distintos contextos mediante el uso de medios, códigos y herramientas apropiados para el desarrollo de algunos temas, estableciendo una postura personal sobre los temas abordados e identificando su relevancia general en su formación, considerando otros puntos de vista de manera crítica y reflexiva, y manteniendo relaciones interpersonales positivas con sus maestros y compañeros de grupo; mostrando una actitud respetuosa hacia la interculturalidad y la diversidad de creencias, valores, ideas y prácticas sociales; desarrollando habilidades matemáticas; desarrollando innovaciones y proponiendo soluciones a problemas a partir de métodos establecidos en este campo específico de la programación.

5. Orientaciones didácticas y estrategias de aprendizaje por unidad

Unidad I:

Manejo de componentes y estructuras básicas de programación orientada a objetos

Orientaciones didácticas (Dirigidas al docente)

En esta unidad el alumno desarrolla las competencias relativas a manejar los componentes y estructuras básicas de la programación orientada a objetos para resolver problemas cotidianos de manera eficiente. Asimismo, se desarrollan las competencias genéricas aplicables de manera natural a las competencias profesionales expresadas en los Resultados de Aprendizaje (RA), con el fin de promover una formación integral en el alumno, por lo que, durante todo el módulo, se fomenta:

- La autonomía, responsabilidad y cuidado de sí mismo, mediante el autoconocimiento que cada alumno va desarrollando, tanto de sus cualidades, como de las áreas en que debe trabajar para su reforzamiento, determinando las acciones de corto, mediano y largo plazo, necesarias para la consecución de los objetivos definidos, considerando los factores sociales, económicos y personales que pueden influir positiva o negativamente en los objetivos contemplados para planear, elegir alternativas y administrar los recursos con los que cuenta.
- Que el alumno proponga soluciones a problemas reales o hipotéticos, con base en actividades de búsqueda de información objetiva y veraz, aplicación de lo aprendido, e innovación en los métodos establecidos. Asimismo, se promueve el análisis crítico y fundamentado.
- El interés y el respeto por la diversidad cultural en todas sus manifestaciones y que el alumno conozca puntos de vista diferentes sobre asuntos de interés público y personal, como condición para conformar el criterio personal de manera libre y sustentada.
- El compromiso con el respeto a la persona, sin distinción de género, y la promoción de la igualdad de oportunidades para hombres y mujeres, asumiendo el alumno el papel de agente de cambio en el proceso de apertura de espacios de participación social y laboral de los que tradicionalmente se ha excluido al género femenino.
- Que el alumno sea capaz de automotivarse en el logro de metas personales y académicas, de desarrollar la capacidad para regular y manejar sus propios impulsos y necesidades, asumir sus propios sentimientos y emociones y encauzarlos positivamente.
- Que sea capaz de continuar aprendiendo de manera cada vez más eficaz y autónoma de acuerdo a los propios objetivos y necesidades, lo que implica aprender a autorregular su proceso de aprendizaje y a resolver diversas problemáticas de la vida académica y profesional, realizando de manera sistemática la planificación de las actividades de aprendizaje, la regulación de su proceso de aprendizaje y la evaluación de los

Unidad I:

Manejo de componentes y estructuras básicas de programación orientada a objetos

Orientaciones didácticas (Dirigidas al docente)

resultados obtenidos tras la aplicación de la estrategia seleccionada.

- Que desarrolle capacidades para establecer una comunicación asertiva y efectiva, en diversos contextos, así como para identificar canales alternos y plurales que diversifiquen la obtención de la información y los enfoques con que ésta es tratada, utilizando una segunda lengua en situaciones cotidianas y en la consulta e interpretación de documentos técnicos.
- Que aprenda a desempeñarse en situaciones de aprendizaje cooperativo y colaborativo, interactuando y trabajando para el logro de los objetivos y metas de aprendizaje del grupo, lo que contribuye también al desarrollo personal y social del alumno.
- Que participe activamente en la democracia, traducida en una mayor equidad en diversos ámbitos sociales y profesionales de su entorno. Todo ello con capacidad de tolerancia y flexibilidad de criterio para alcanzar consensos.
- Que incorpore medidas de seguridad e higiene en el desempeño de sus actividades profesionales.
- Que adquiera el compromiso social de sustentabilidad, aplicable más allá de lo relativo al medio ambiente, orientándose a la satisfacción de las necesidades actuales, sin perjuicio de las futuras generaciones en el plano social, tecnológico, económico, cultural y cualquier otro que se relacione con la preservación y bienestar de la especie humana.
- Que aprenda a minimizar el impacto de sus actividades cotidianas sobre el medio ambiente; consuma responsablemente; se desempeñe con seguridad, calidad y ética en espacios naturales y urbanos; elimine contaminantes o las fuentes de riesgo antes de que se generen, y seleccione y emplee materiales reciclables y biodegradables.
- Que aprenda a movilizar sus recursos personales (conocimientos, habilidades, actitudes y valores) y utilizar estrategias efectivas de aprendizaje continuo para ingresar, mantenerse, desarrollarse y “navegar” en el mundo del trabajo, a lo largo de su trayectoria laboral, ya sea en contextos de trabajo dependientes como independientes.

Asimismo, el docente diseña actividades que promueven el desarrollo y formación integral del alumno y realiza el acompañamiento en la identificación de problemas y situaciones que son una barrera en su aprendizaje y en la construcción de competencias, con la finalidad de ayudarlo a descubrir su potencial, enfrentar y superar los retos de la vida utilizando, incrementando la confianza en sí mismo, consolidando su firmeza en la consecución de sus metas.

Para el efecto, se aplicarán las técnicas: método de proyecto, complementado con presentaciones grupales para promover la discusión y análisis, análisis de lecturas en corrillos, análisis de problemas e interrogación, con un enfoque de aprendizaje significativo y colaborativo, consulta de información con proveedores y fabricantes manejando información actualizada y contactando con posibles fuentes de trabajo futuras.

Unidad I:

Manejo de componentes y estructuras básicas de programación orientada a objetos

Orientaciones didácticas (Dirigidas al docente)

Actividades sugeridas:

1. Inicia la sesión presentándose ante el grupo. Da una introducción general del módulo y analiza en conjunto los resultados de aprendizaje que se pretenden lograr. Establece la forma de trabajo en clase y explica cómo se llevarán a cabo las actividades de evaluación, considerando las rúbricas correspondientes. Explica brevemente las competencias a adquirir en el módulo y la forma en que contribuirán a las competencias contenidas en el perfil de egreso, así como su relación con módulos pasados, presentes y futuros de la carrera. Asimismo, invita a los alumnos a practicar los valores de respeto, dignidad, la no-violencia, la responsabilidad, el orden, la limpieza y el trabajo en equipo en todas sus actividades y relaciones que establezcan.
2. Realiza una evaluación diagnóstica sobre conceptos de diseño de algoritmos y programación estructurada, para identificar los aspectos que son necesarios reforzar; Solicita a los alumnos su compromiso para estudiar lo necesario para alcanzar la competencia del módulo; Orienta al grupo para presentar sus expectativas del curso y así definir sus metas de aprendizaje y estrategias para alcanzarlas, haciendo uso de sus habilidades, valores y fortalezas.
3. Solicita al alumno el desarrollo de un proyecto que dé solución a un problema de su entorno, de acuerdo con lo requerido para validar la acreditación del módulo recabando las evidencias y considerando el siguiente procedimiento para su elaboración:
4. Determina la finalidad del Proyecto a desarrollar de manera que el alumno sepa lo que se espera de él antes de comenzar a preparar sus productos de evidencia.
5. Acuerda con los alumnos y con base en el programa de estudios, los productos académicos que serán entregados como proyecto, incluyendo los productos generados durante las actividades de evaluación y las rúbricas correspondientes.
6. Las evidencias recopiladas han de establecer una correspondencia entre el trabajo a lo largo del desarrollo del módulo y sus experiencias de aprendizaje.
7. Las evidencias deberán estar relacionadas entre sí, de manera que se observe una lógica en el proyecto
8. La elaboración de los productos académicos puede solicitarse por segunda vez, si en la primera presentan deficiencias, con otro valor en el puntaje que será menor al establecido de inicio.
9. Organiza al grupo en equipos de trabajo y solicita realicen una investigación vía internet sobre la Introducción a la Programación Orientada a Objetos, considerando en que consiste y cuáles son sus fundamentos, solicita que complementen dicha investigación organizando una lluvia de ideas para intercambiar opiniones con el grupo
10. Explica mediante esquemas, los Conceptos fundamentales de la Programación Orientada a Objetos, Identifica las ideas clave de la exposición objeto, clase y herencia hace conclusiones con el grupo a partir de ellas.
11. Expone mediante el proyector de acetatos, las Características de los Lenguajes orientados a objetos, solicitando un resumen de lo expuesto y su complemento mediante una investigación documental y de campo en la que obtengan una tabla comparativa de cuando menos 6 lenguajes de

Unidad I:	Manejo de componentes y estructuras básicas de programación orientada a objetos
Orientaciones didácticas (Dirigidas al docente)	
<p>Programación Orientados a Objetos con sus principales características.</p> <p>12. Organiza una investigación de campo individual para obtener información sobre la representación de la programación Orientada a Objetos en las computadoras: clases, relaciones entre objetos, tipos de datos, variables, constantes, operadores, punteros y referencias. Fomenta la equidad de género y la igualdad en la designación de actividades de desarrollo profesional</p> <p>13. Explica la Estructura de un programa en un Lenguaje Orientado a Objetos, Comentarios, Función principal, Archivos de cabecera, Cuerpo del programa, Entrada y salida de datos y Sentencias de control, solicita la repetición del procedimiento para u una serie de 3 ejercicios con distintos tipos de solución. Da orientaciones sobre la toma decisiones para proponer la solución a los ejercicios.</p> <p>14. Utiliza diferentes recursos didácticos (presentación en power point, presentación de la pantalla directamente en el pizarrón, rotafolio, rompecabezas, etiquetas adhesivas un cuarto de carta), en el Manejo de las clases en la programación, manejo de las clases en la programación, dirigiéndola a conceptos, componentes, atributos, métodos, tipos de funciones, características de constructores, características destructores, variables de clases, con abundantes ejemplos que permitan a los alumnos apoderarse de los conceptos y su aplicación. Solicita a los alumnos elaboren en equipo ejemplos de aplicación de los conceptos manejados.</p> <p>15. Orienta y apoya la realización de la Práctica 1 “Elabora un programa básico describiendo los componentes y características de la programación orientada a objetos”, correspondiente a la actividad de evaluación 1.2.1. En la rúbrica correspondiente se incluye una Coevaluación.</p> <p>16. Aplica un cuestionario escrito, con el propósito de afirmar el conocimiento del alumno en el Manejo de componentes y estructuras básicas de programación orientada a objetos; la descripción de sus principales características, los fundamentos de la programación orientada a objetos, considerando la estructura de programas bajo este ambiente y solicita a los alumnos que respondan de manera individual las preguntas y posteriormente circulen por el aula buscando compañeros que hayan contestado las preguntas que ellos no pudieron para completar la información que les falte.</p>	

Estrategias de aprendizaje (dirigidas al alumno)	Recursos académicos
<p>El alumno:</p> <ul style="list-style-type: none"> Expone sus expectativas del curso y analiza las actividades de aprendizaje, los criterios de evaluación y el método de aprendizaje. Plantea sus dudas y toma nota sobre los puntos explicados por el docente. Contesta la evaluación diagnóstica sobre conceptos de diseño de algoritmos y 	<ul style="list-style-type: none"> 1 Computadora con procesador de texto y software de presentación para docente. 1 Computadora cuando menos por cada 3 alumnos. Lenguaje C++. Compilador Visual C++ (Windows).

Estrategias de aprendizaje (dirigidas al alumno)	Recursos académicos
<p>programación estructurada. Se compromete a reforzar los aspectos importantes del tema y a adquirir los conocimientos mínimos necesarios para cursar el módulo, por lo que establece metas y estrategias para su logro. Asimismo, se propone practicar los valores de respeto, dignidad, la no-violencia, la responsabilidad, el orden, la limpieza y el trabajo en equipo en todas sus actividades y relaciones que establezcan.</p> <ul style="list-style-type: none"> • Contesta la evaluación diagnóstica sobre conceptos de diseño de algoritmos y programación estructurada. Se compromete a reforzar los aspectos importantes del tema y a adquirir los conocimientos mínimos necesarios para cursar el módulo, por lo que establece metas y estrategias para su logro, haciendo uso de sus habilidades, valores y fortalezas. • Pone atención a la exposición y explicación del docente mostrando una actitud de respeto e interés por su aprendizaje, realiza una investigación vía internet sobre la Introducción a la Programación Orientada a Objetos, considerando en que consiste y cuáles son sus fundamentos, participa activamente en la lluvia de ideas para intercambiar opiniones con sus compañeros. Al tiempo que reconoce que la diversidad tiene lugar en un espacio democrático de igualdad de dignidad y derechos de todos sus compañeros, y rechaza toda forma de discriminación. • Analiza en los esquemas expuestos por el docente, los Conceptos fundamentales de la Programación Orientada a Objetos, Identifica las ideas clave de la exposición objeto, clase y herencia hace conclusiones con sus compañeros. Asume una actitud constructiva, congruente con los conocimientos y habilidades con los que cuenta dentro de distintos equipos de trabajo en los que participa de forma activa. • Toma nota sobre los puntos relevantes de la Exposición del docente acerca de las Características de los Lenguajes orientados a objetos, elaborando un resumen de lo expuesto y realiza una investigación documental y de campo de al menos 6 lenguajes de Programación Orientados a Objetos con sus principales características. Ordena información de acuerdo a categorías, jerarquías y relaciones respecto al tema. • Realiza una investigación de campo individual para obtener información sobre la representación de la programación Orientada a Objetos en las computadoras: clases, relaciones entre objetos, tipos de datos, variables, constantes, operadores, punteros y referencias. Fomenta la equidad de género y la igualdad en la realización de actividades • Atiende la explicación del docente acerca de la Estructura de un programa en un Lenguaje 	<ul style="list-style-type: none"> • Poseidón para UML (software de modelado de programas), Community edition. • Librerías: eclipse, OpenGL. • Cañón de proyección. • Pantalla. • Escritorios. • Manuales técnicos de programas y librería. • Balagurusamy, E. <u>Programación orientada a objetos con C++</u>. 3ª Ed. McGraw Hill, Madrid, 2006. • Cáceres González, Abdiel. <u>Notas Programación orientada a objetos</u>. CINVESTAV, IPN, México, 2004. • Fernández Sastre, Sergio. <u>Fundamentos de diseño y programación orientada a objetos</u>. Mc Graw Hill; México, 2006. • Larman, C., <u>UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado</u>, 2a edición, Pearson Educación 2002 • Marina, José L. <u>Curso de programación C++</u>, http://www.jlmarina.net/metodos2/ (10/07/2015) • Meyer, B., <u>Construcción de software orientado a objetos</u>, 2a Ed., Prentice-Hall, 1999. • Rumbaugh, J; Jacobson, I; Boch, G. <u>El lenguaje unificado de modelado. Manual de referencia</u>. Addison Wesley, España, 2000. • Conceptos y prácticas de programación orientada a objetos, Disponible en: http://dis.um.es/~bmoros/privado/apuntes/Curso09-10/POO1-0910.pdf, (10/07/2015). • Enlaces a páginas de programación orientada a objetos, Disponible en: http://www.arrakis.es/~devis/oo.html.

Estrategias de aprendizaje (dirigidas al alumno)	Recursos académicos
<p>Orientado a Objetos, Comentarios, Función principal, Archivos de cabecera, Cuerpo del programa, Entrada y salida de datos y Sentencias de control. Repite el procedimiento para una serie de 3 ejercicios con distintos tipos de solución. Solicita apoyo para proponer la solución a los ejercicios.</p> <ul style="list-style-type: none">• Escucha con atención al docente en su exposición del Manejo de las clases en la programación, dirigida a conceptos, componentes, atributos, métodos, tipos de funciones, características de constructores, características destructores, variables de clases. Toma las notas pertinentes que le permitan apoderarse de los conceptos y su aplicación. Elabora en equipo los ejemplos de aplicación solicitados.• Realiza la Práctica 1 “Elabora un programa básico describiendo los componentes y características de la programación orientada a objetos”, correspondiente a la actividad de evaluación 1.2.1 y participa en la actividad de Coevaluación.• Contesta el cuestionario escrito, con el propósito de afirmar sus conocimientos en el Manejo de componentes y estructuras básicas de programación orientada a objetos; la descripción de sus principales características, los fundamentos de la programación orientada a objetos, considerando la estructura de programas bajo este ambiente. Responde de manera individual las preguntas y posteriormente circula por el aula buscando compañeros que hayan contestado las preguntas que el no pudo para completar la información	<p>(10/07/2015).</p> <ul style="list-style-type: none">• Poseidón software de modelado de programas para usar UML, Disponible en: http://www.gentleware.com/ (10/07/2015).

Unidad II:	Creación de programas orientados a objetos
Orientaciones didácticas (Dirigidas al docente)	
<p>Esta unidad está enfocada a que el alumno adquiera la competencia en el desarrollo de programas utilizando los fundamentos, estructuras, modelos de diseño, diagramas y técnicas básicas de la programación orientada a objetos en entorno de lenguaje orientado a objetos, como el C++, así como refuerza las competencias genéricas de trabajo en equipo, análisis y solución de problemas, promueve los valores: responsabilidad, disciplina, tolerancia y liderazgo, apoyando al alumno en su desarrollo integral, ayudándolo a identificar problemas que le impiden la aplicación de competencias desarrolladas con anterioridad y la adquisición de nuevas competencias, a ayudarlo a que identifique su potencial y a acompañarlos en la consecución de sus metas, fortaleciendo su seguridad y confianza en el mismo con sus logros.</p> <p>Para esto, en la presente unidad se emplearán las técnicas participativo – vivenciales, el método de proyectos y estudio de casos, bajo el enfoque de aprendizaje significativo y colaborativo, descritos en el apartado 3 de la presente guía.</p> <p>Actividades sugeridas:</p> <ol style="list-style-type: none">1. Describe el manejo de objetos con un lenguaje de programación, específicamente las propiedades generales del lenguaje C++, a través de una presentación en power point su proceso de instalación, el inicio de la escritura de un programa, librerías requeridas y sintaxis así como su invocación, el uso de Operadores, diseño de objetos, creación y destrucción de objetos, mensajes entre objetos, Interfases, Encapsulamiento, Genericidad, Atributo, Variables, funciones miembro, uso de <nombre>.ccp y uso de <nombre>.h. explicando la función de los constructores. y destructores; con el fin de desarrollar cuando menos 5 ejemplos de la aplicación de estos conceptos, y presentarlos para su discusión en el aula.2. Contacta con profesionales que impartan el módulo u otros módulos de la misma carrera y que funjan como docentes de otros planteles; genera un Blog personal para compartir materiales y recursos.3. Realiza una demostración práctica en el laboratorio en la que incluye el uso del compilador del Lenguaje C++, codificación del programa, compilación del programa fuente, depuración de los posibles errores aparecidos, linkear o enlazar los componentes del programa objeto, uso de teclas rápidas en el compilador, ejecución del programa; así como sus generalidades; elabora varios ejemplos de elaboración de un programa que presente un texto en la pantalla, la suma, resta, producto y cociente de números.4. Elabora ejemplos sobre el manejo de las propiedades de la programación orientada a objetos en situaciones reales: aplicando los conceptos de Herencia y Polimorfismo con aplicación en el lenguaje orientado a objetos C++, los cuales expondrá en el aula con pantalla proyectada, para su discusión y retroalimentación.5. Divide al grupo en equipos de 4 integrantes para los que asigna una investigación vía internet sobre el manejo del análisis orientado a objetos, comprensión del problema, elaboración de las especificaciones de requerimientos del usuario y del software, identificación de los objetos a utilizar y sus atributos incluyendo los servicios que deberá ofrecer cada objeto (interfase); solicitando a los alumnos el desarrollo de los mismos en un diseño	

Unidad II:

Creación de programas orientados a objetos

Orientaciones didácticas (Dirigidas al docente)

- de tríptico ó díptico para su posterior intercambio entre equipos, fomentando la convivencia entre ellos.
6. Presenta un diagrama conceptual sobre las propiedades, estructura, procesos, análisis, manejo de la metodología UML y pasos en el desarrollo de un programa orientado a objetos, con la finalidad de ubicar al alumno en el contexto de la representación de los componentes de un programa a través de modelos (diagramas) y particularizando inicialmente en el manejo de las propiedades del programa. Asigna al alumno elabore 10 casos en el manejo de las características y propiedades de un programa orientado a objetos.
 7. Coordina una lluvia de ideas para realizar el análisis los fundamentos de los programas básicos con programación orientada a objetos en especial los realizados en lenguaje C++, los objetos y clases, asegurándose de que todos los integrantes de los equipos participen en la actividad.
 8. **Orienta y apoya la realización de la Práctica 2 “Desarrolla un programa en lenguaje C++, manejando las propiedades de objetos y clases”, correspondiente a la actividad de evaluación 2.1.1.**
 9. Desarrolla varios ejemplos proyectados en pantalla, sobre el manejo de la estructura de un programa en el entorno del lenguaje C++, haciendo participar alternadamente a los alumnos. Asigna actividades de práctica en el manejo de la estructura para situaciones reales, los cuales presentarán en el aula para discusión y retroalimentación.
 10. Ejemplifica con varios casos el manejo del análisis orientado a objetos, enfocado al desarrollo de programas en lenguaje C++, integrando los componentes, estructuras, modelos de diseño y representación de la programación orientada a objetos, compilados y en ejecución
 11. Asigna al alumno la realización de cuando menos 10 casos en la aplicación de los conceptos discutidos para el análisis, con la finalidad que los discutan en el aula y reciba retroalimentación.
 12. Realiza ejemplos de aplicación de todos los conceptos, metodología, sintaxis en el uso del lenguaje C++, relacionando la metodología orientada a objetos con la utilización del lenguaje C++ en el desarrollo de programas con este enfoque, ejecutando los programas y revisando los resultados obtenidos. Asigna un proyecto para la solución de un problema de la vida cotidiana, en el que los alumnos, organizados en equipo, desarrollen un programa orientado a objetos en el entorno de lenguaje C++, su ejecución y evaluación de los resultados obtenidos. Promueve una lluvia de ideas sobre los pasos a realizar en el diseño de un programa orientado a objetos; cerrando este tema reforzando los conceptos que corresponden a los pasos en un diseño de esta índole.
 13. **Orienta y apoya la realización de la Práctica 3 “Desarrolla un programa de aplicación final, empleando la metodología y técnica de un Lenguaje de programación orientada a objetos”, correspondiente a la actividad de evaluación 2.2.1.**

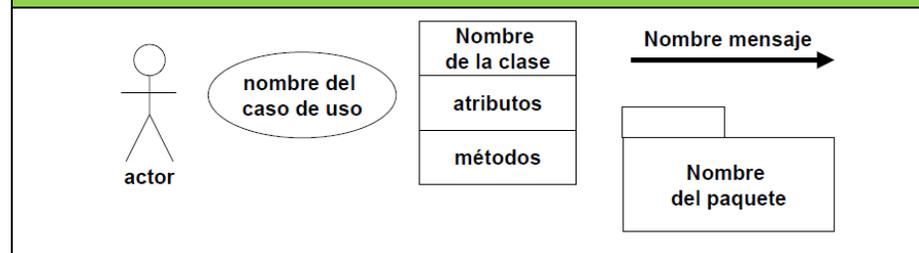
Estrategias de aprendizaje (dirigidas al alumno)	Recursos académicos
<p>El alumno:</p> <ul style="list-style-type: none"> • Atiende la demostración del docente acerca del manejo de objetos con un lenguaje de programación, atendiendo a su proceso de instalación, el inicio de la escritura de un programa, librerías requeridas y sintaxis así como su invocación, el uso de Operadores, diseño de objetos, creación y destrucción de objetos, mensajes entre objetos, Interfases, Encapsulamiento, Genericidad, Atributo, Variables, funciones miembro, uso de <nombre>.ccp y uso de <nombre>.h. la función de constructores y destructores; y desarrolla 5 ejemplos de la aplicación de estos conceptos, presentándolos para su discusión en el aula. • Se registra como amigo del docente y entra a su blog personal para acceder a la información depositada referente programas de aplicación final. Registra su visita en dicho sitio para evidenciar el uso de la herramienta. • Observa la demostración práctica hecha por el docente acerca de la elaboración de ejemplos de elaboración de un programa que presente un texto en la pantalla, la suma, resta, producto y cociente de números. • Atiende la explicación de ejemplos sobre el manejo de las propiedades de la programación orientada a objetos en situaciones reales: conceptos de Herencia y Polimorfismo con aplicación en el lenguaje orientado a objetos C++, realiza preguntas y resuelve las dudas presentadas. • Se integra en equipos de 4 integrantes realizando una investigación vía internet sobre el manejo del análisis orientado a objetos, comprensión del problema, elaboración de las especificaciones de requerimientos del usuario y del software, identificación de los objetos a utilizar y sus atributos incluyendo los servicios que deberá ofrecer cada objeto (interfase); elaborando un diseño de tríptico ó díptico para su posterior intercambio con sus compañeros de los otros equipos. • Analiza el diagrama conceptual presentado por el docente, sobre las propiedades, estructura, procesos, análisis, manejo de la metodología UML y pasos en el desarrollo de un programa orientado a objetos, ubicándose en el contexto de la representación de los componentes de un programa a través de modelos (diagramas) y el manejo de las propiedades de éste. Elabora 10 casos en los que incluye la aplicación de los conceptos 	<ul style="list-style-type: none"> • 1 Computadora con procesador de texto y software de presentación para docente. • 1 Computadora cuando menos por cada 3 alumnos. • Lenguaje C++. • Compilador Visual C++. • Poseidón para UML (software de modelado de programas) • Librerías: eclipse, OpenGL. • Cañón de proyección. • Pantalla. • Escritorios. • Manuales técnicos de programas y librería. • Balagurusamy, E. <u>Programación orientada a objetos con C++</u>. 3ª Ed. Mc Graw Hill, Madrid, 2006. • Cáceres González, Abdiel. <u>Notas Programación orientada a objetos</u>. CINVESTAV, IPN, México, 2004. • Fernández Sastre, Sergio. <u>Fundamentos de diseño y programación orientada a objetos</u>. Mc Graw Hill; México, 2006. • Larman, C., UML y Patrones. <u>Una introducción al análisis y diseño orientado a objetos y al proceso unificado</u>, 2a edición, Pearson Educación 2002 • Meyer, B., <u>Construcción de software orientado a objetos</u>, 2a Ed., Prentice-Hall, 1999. • Rumbaugh, J; Jacobson, I; Boch, G. <u>El lenguaje unificado de modelado. Manual de referencia</u>. Addison Wesley, España, 2000. • Conceptos y prácticas de programación orientada a objetos, Disponible en: http://dis.um.es/~bmoros/privado/apuntes/Curso09

Estrategias de aprendizaje (dirigidas al alumno)	Recursos académicos
<p>del manejo de las características y propiedades de un programa orientado a objetos.</p> <ul style="list-style-type: none"> Participa activamente en una lluvia de ideas realizando el análisis los fundamentos de los programas básicos con programación orientada a objetos en especial los realizados en lenguaje C++, los objetos y clases. Realiza la Práctica 2 “Desarrolla un programa en lenguaje C++, manejando las propiedades de objetos y clases”, correspondiente a la actividad de evaluación 2.1.1. Analiza los ejemplos proyectados en pantalla, sobre el manejo de la estructura de un programa en el entorno del lenguaje C++, participa alternadamente con sus compañeros. Realiza actividades de práctica en el manejo de la estructura para situaciones reales, discute y retroalimenta. Atiende los ejemplos acerca del manejo del análisis orientado a objetos, enfocado al desarrollo de programas en lenguaje C++, integra los componentes, estructuras, modelos de diseño y representación de la programación orientada a objetos, compilados y en ejecución Realiza 10 ejercicios sobre la aplicación de los conceptos discutidos. Observa los ejemplos de aplicación de todos los conceptos, metodología, sintaxis en el uso del lenguaje C++, realizando un proyecto para la solución de un problema de la vida cotidiana, en el que organizado en equipo, ejecuta y evalúa los resultados obtenidos. Participa activamente en una lluvia de ideas sobre los pasos a realizar en el diseño de un programa orientado a objetos. Realiza la Práctica 3 “Desarrolla un programa de aplicación final, empleando la metodología y técnica de un Lenguaje de programación orientada a objetos”, correspondiente a la actividad de evaluación 2.2.1. 	<p>-10/POO1-0910.pdf, (10/07/2015).</p> <ul style="list-style-type: none"> Enlaces a páginas de programación orientada a objetos, Disponible en: http://www.arrakis.es/~devis/oo.html, [consulta en 14 abril2010]. Librerías de C++, Disponible en: http://www.cplusplus.com/reference/ (10/07/2015)

6. Prácticas/Ejercicios /Problemas/Actividades

Unidad de aprendizaje:	Manejo de componentes y estructuras básicas de programación orientada a objetos	Número:	1
Práctica:	Elabora un programa básico describiendo los componentes y características de la programación orientada a objetos	Número:	1
Propósito de la práctica:	Identificar las propiedades y características de los objetos y clases, como unidades fundamentales de la programación orientada a objetos.		
Escenario:	Taller o laboratorio	Duración	4 horas
Materiales, herramientas, instrumental, maquinaria y equipo		Desempeños	
<ul style="list-style-type: none"> • 1 Computadora con procesador de texto y software de presentación para docente. • 1 Computadora cuando menos por cada 3 alumnos. • Poseidón comunista edition (software para utilizar la metodología UML). • Cañón de proyección. • Pantalla. • Escritorios. • Manuales técnicos de programas y librerías. 		<ul style="list-style-type: none"> • Aplica las medidas de seguridad e higiene durante el desarrollo de la práctica. • Prepara el equipo, las herramientas y los materiales a utilizar. • Integra equipos de trabajo de 4 participantes: <p>Identificación del entorno de los objetos en Programación Orientada a Objetos</p> <ul style="list-style-type: none"> • Identifica en la Fig. 1 que tipo de elemento representa, considerando que existen: estructurales, de comportamiento y de agrupación. • Describe las partes del elemento, proporcionando referencias de uso. • Desarrolla 5 ejemplos de la forma de utilizar la estructura de la Fig. 1. 	

Fig. 1. Tipos de elementos, éste ¿a cuál corresponde?



- Utiliza la representación análoga a la Fig. 1, mostrando un elemento que represente mensajes.
- Utiliza la representación análoga a la Fig. 1, mostrando un elemento que represente paquetes.
- Identifica el tipo de relación presente entre los elementos de la Fig. 2, Fig. 3, Fig. 4, Fig. 5, considerando que existen: generalización, dependencia, asociación.

Fig. 2. Relación

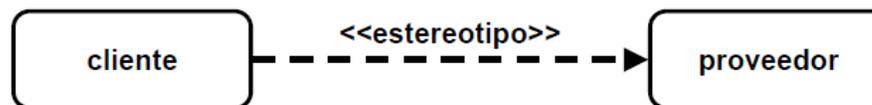


Fig. 3. Tipos de relación

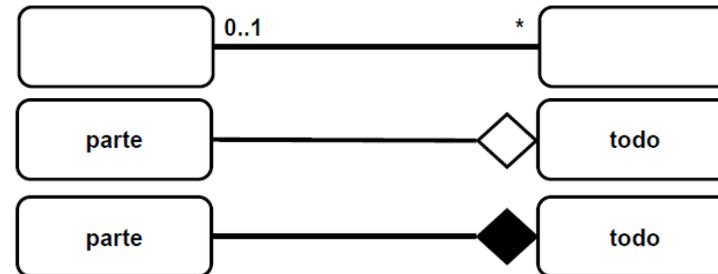


Fig. 4. Relación

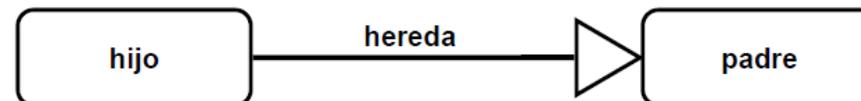
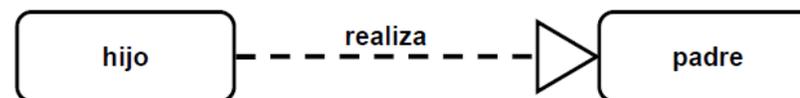
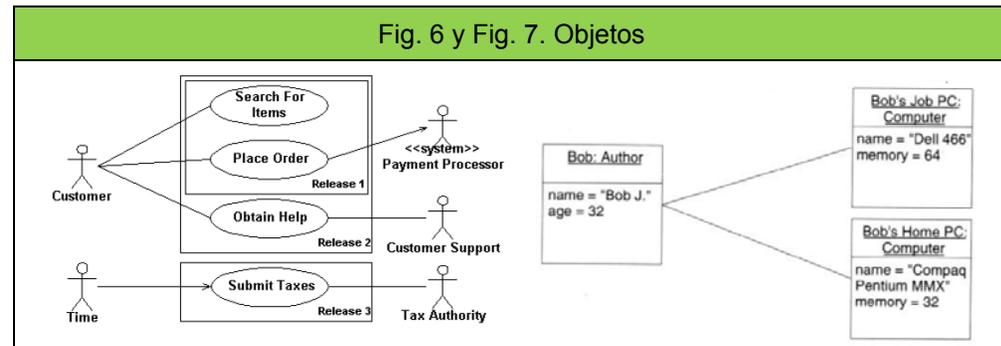


Fig. 5. Relación



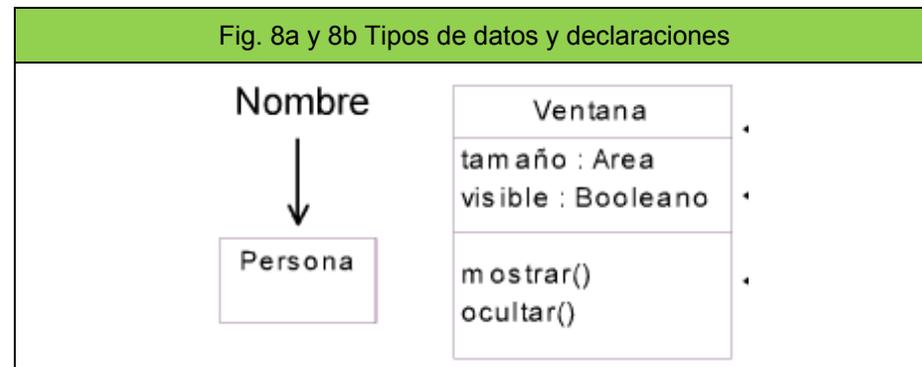
- Desarrolla 5 ejemplos de la forma en que se utilizan cada tipo de relación entre elementos.

- Describe que nos representan las Fig. 6 y Fig. 7; presenta dos ejemplos de su uso.



Identificación del entorno de los clases en Programación Orientada a Objetos

- Describe el concepto que representa la Fig. 8a y 8b, así como cada uno de sus componentes.
- Desarrolla 5 ejemplos de aplicación de la Fig. 8a y 8b.



	<div data-bbox="1066 282 1684 841" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"><p style="text-align: center;">Ventana</p><p>+ tamaño : Area = (100,100) # visible : Booleano + tamaño-defecto : Rectángulo # tamaño-máximo : Rectángulo - Xptr : Ventana</p><hr/><p>+ mostrar() + ocultar() + crear()</p></div> <ul style="list-style-type: none">• Utiliza la estructura de la Fig. 9, para representar cualquier otra situación de la vida cotidiana que requiera ser resuelta a través de programación orientada a objetos. Identifica componentes del programa de la Fig. 9, describe sus elementos, en la cual identifica metodología de representación de programas.
--	--

Fig. 9 Elementos y componentes de un programa de alguna situación real

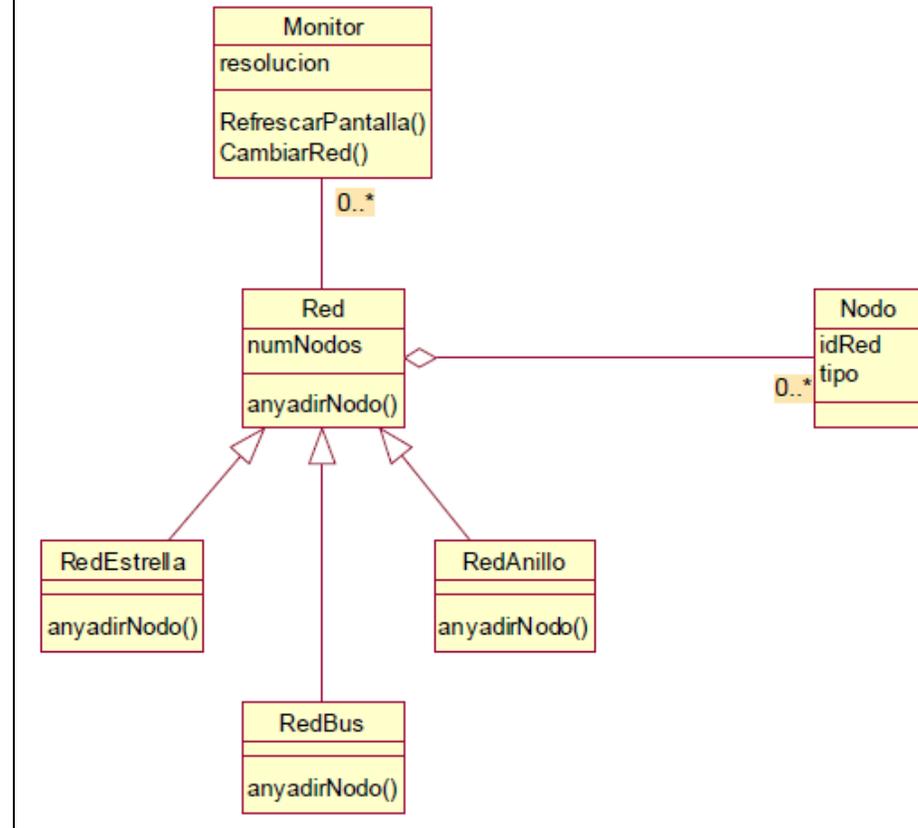
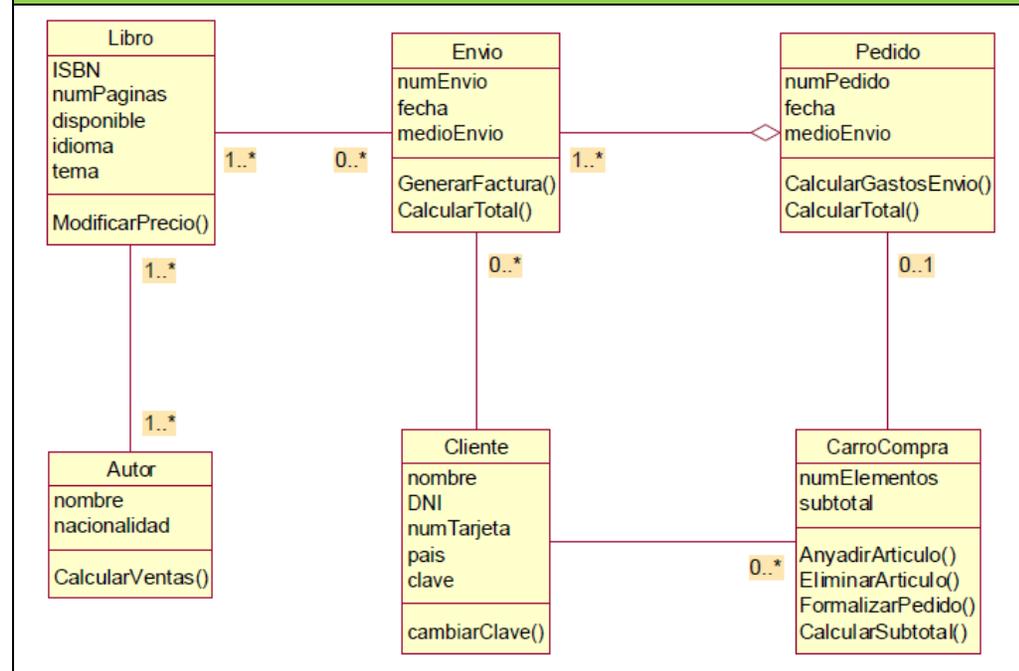


Fig. 9 Elementos y Componentes de un programa de alguna situación real



Presentación de programa con la identificación de componentes.

- Presenta programa señalando los componentes y sus características.
- Elabora reporte de resultados y conclusiones.

Unidad de aprendizaje:	Creación de programas orientados a objetos	Número:	2		
Práctica:	Desarrolla un programa en lenguaje C++, manejando las propiedades de objetos y clases	Número:	2		
Propósito de la práctica:	Manejar las propiedades de objetos y clases, así como su aplicación en un lenguaje orientado a objetos.				
Escenario:	Taller o laboratorio	Duración	6 horas		
Materiales, herramientas, instrumental, maquinaria y equipo		Desempeños			
<ul style="list-style-type: none"> • 1 Computadora con procesador de texto y software de presentación para docente. • 1 Computadora cuando menos por cada 3 alumnos. • Lenguaje C++. • Compilador Visual C++. • Librerías: eclipse, OpenGL. • Cañón de proyección. • Pantalla. • Escritorios. • Manuales técnicos de programas y librería. 		<ul style="list-style-type: none"> • Integra equipos de trabajo de 3 participantes: • Aplica las medidas de seguridad e higiene en el desarrollo de la práctica. <p>Identificación elementos, instrucciones y operadores en C++.</p> <ul style="list-style-type: none"> • Describe la sintaxis y forma en que operan las siguientes instrucciones, el resultado de la instrucción y desarrolla 5 ejemplos de aplicación de las mismas: 			
		<table border="1"> <thead> <tr> <th>Programa de ejemplo</th> </tr> </thead> <tbody> <tr> <td> <pre>#include <iostream> // librerías using namespace std; int main() // inicio del programa de ejemplo { cout << "Estamos usando el Lenguaje C++ en una práctica con el fin de ejemplificar la programación orientada a objetos.\n"; // Instrucción C++ return 0; } // Fin del programa de ejemplo</pre> </td> </tr> </tbody> </table>		Programa de ejemplo	<pre>#include <iostream> // librerías using namespace std; int main() // inicio del programa de ejemplo { cout << "Estamos usando el Lenguaje C++ en una práctica con el fin de ejemplificar la programación orientada a objetos.\n"; // Instrucción C++ return 0; } // Fin del programa de ejemplo</pre>
Programa de ejemplo					
<pre>#include <iostream> // librerías using namespace std; int main() // inicio del programa de ejemplo { cout << "Estamos usando el Lenguaje C++ en una práctica con el fin de ejemplificar la programación orientada a objetos.\n"; // Instrucción C++ return 0; } // Fin del programa de ejemplo</pre>					

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<div data-bbox="905 358 1841 1138" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center; background-color: #92d050; margin: 0;">Programa de ejemplo 2</p> <pre style="margin: 0;">#include <iostream> // librerías using namespace std; int main() // inicio del programa de ejemplo { Float numero1, numero2, // declaración tipo de datos y // variables suma, media; cout << "Introduzca dos números: ."; // Petición de valores cin >>numero1; // Lee números cin >>numero1; // desde teclado suma =numero1 + numero2; media = suma / 2 cout << "Suma = " << suma << "\n "; cout << "Media = " << media << "\n "; return 0; } // Fin del programa de ejemplo.</pre> </div> <ul style="list-style-type: none"> Describe línea por línea lo que realiza el programa representado en Fig. 1 y Fig. 2, así como los resultados que entrega. Identifica la representación de clases en el programa. Describe elemento objetos y los resultados que entrega el programa.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños	
	<p>Fig. 1. Programa básico</p> <pre data-bbox="829 414 1373 1323"> #include <iostream> using namespace std; class persona { char nombre[30]; int age; public: void getDatos(void); void mostrar(void); }; void persona::getDatos(void) { cout << "Introduzca el nombre: "; cin >> nombre; cout << "Introduza edad: "; cin >> edad; } void persona::mostrar(void) { cout << "\nNombre: " << nombre; cout << "\nEdad: " << edad; } int main () { person p; p.getDatos(); p.mostrar(); return 0; } </pre>	<p>Fig. 2. Objetos y los resultados</p> <pre data-bbox="1373 414 1917 1323"> class X { inti_; public: X(inti); intf() const; }; X::X(inti):i_(i){} intX::f()const {returni_;} intmain() { Xx1(10); constXx2(20); x1.f(); x2.f(); } </pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>Identificación de la aplicación de la herencia y el polimorfismo</p> <ul style="list-style-type: none"> Identifica cada elemento del programa, la función que realiza en él y los resultados que entrega, Identifica en que parte del programa se maneja la herencia. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center; background-color: #92d050; margin: -10px -10px 10px -10px;">Herencia y Polimorfismo</p> <pre> class Coche { private: Rueda a_[4]; Motor motor_; Direccion dir_; public: Rueda(): motor_(0), dir_(0){}; arrancar() motor_.encender(); dir_.centrar(); } int main() { Coche c1; c1.arrancar();//Bien. c1.a_[0].inflar();//Mal.Privado } </pre> <hr/> <pre> class Persona { private: char* nif; int edad; } </pre> </div>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre> char* nombre,*apellidos; public: Persona(char*,int=0,char*,char*); Persona&operator=(Persona&); ~Persona(); voidmedad(int); voidmnombre(char*); char*mnombre(); voidmostrar(); char*nombreCompleto(); voidfelizCumple(); voidleer(); }; /* Polimorfismo */ voidPersona::felizcumple() { edad++; cout<<"elicitades!!" mostrar(); } intmain() { Alumnoalum("7777R"20, "Ana","uiz"3); alum.felizcumple(); } </pre> <p>Describe todos los elementos, componentes, fundamentos y propiedades de la programación orientada a objetos que se presentan en este programa, así como el resultado que entrega el mismo.</p>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<div data-bbox="905 396 1841 467" style="background-color: #92d050; padding: 5px; text-align: center;"> Componentes, fundamentos y propiedades de la programación orientada a objetos </div> <pre data-bbox="905 467 1841 1360"> #include <iostream.h> /* ***** clase posición ***** */ class posicion { private: int X,Y; public: // característica de lenguaje híbrido void iniciar (int iniciarX, int iniciarY) { X = iniciarX; Y = iniciarY; }; int leerX(); int leerY(); virtual void mensaje(){ cout << "Hola estoy en posicion\n";} }; // fin posición int posicion::leerY() { return Y; } int posicion::leerX() { return X; } /* ***** clase punto ***** */ class punto : public posicion { int visible; public: virtual void mostrar(); virtual void ocultar(); int EsVisible(); virtual void saluda(){ cout << " Hola soy un punto\n";} }; // fin punto void punto::mostrar() { </pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre> visible = 1; }; void punto::ocultar() { visible = 0; }; int punto::EsVisible() { return visible; } /* ***** clase circulo ***** */ class circulo : public punto { int radio; public: void iniciar (int iniciarX, int iniciarY, int IniciaRadio) { punto::iniciar(iniciarX, iniciarY); radio = IniciaRadio; }; void expandir(int ExpandirPor); void comprimir(int ComprimirPor); virtual void saluda(){ cout << " Hola soy un circulo\n";}; }; void circulo::expandir(int ExpandirPor) { radio = radio + ExpandirPor; }; void circulo::comprimir(int ComprimirPor) { radio = radio - ComprimirPor; } /* ***** fin clase circulo ***** */ void main (void) { int Z; circulo posicionA; posicionA.iniciar(16,3,35); posicionA.ocultar(); if (posicionA.EsVisible()) { Z = posicionA.leerX(); </pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre data-bbox="905 305 1841 647">cout << "\n valor visible de X " << Z; posicionA.saluda(); } else cout << "\n valor X no visible " ; }</pre> <p data-bbox="800 699 1272 727">Uso del compilador del Lenguaje C++</p> <ul data-bbox="800 740 1892 841" style="list-style-type: none">• Ejecuta programa fuente, obtiene programa objeto, ejecuta el programa e imprime los resultados para los programas de esta práctica.• Reporta resultados del manejo de los conceptos y resultados de ejecución del programa. <p data-bbox="800 906 1940 964">Manejo de la metodología UML de representación de modelos, en el modelaje y desarrollo de programas</p> <ul data-bbox="800 977 1881 1094" style="list-style-type: none">• Realiza la identificación de los elementos de la metodología UML de representación de modelos, en el modelaje y desarrollo de programas, interpretando sus representaciones gráficas y usos. Sintetiza evidencias obtenidas durante el desarrollo de la actividad, emitiendo conclusiones y formulando preguntas que le permitan aclarar posibles dudas

Unidad de aprendizaje:	Desarrollo de programas utilizando la programación básica orientada a objetos.	Número:	2
Práctica:	Desarrolla un programa de aplicación final, empleando la metodología y técnica de un Lenguaje de programación orientada a objetos	Número:	3
Propósito de la práctica:	Manejar los modelos de la programación orientada a objetos en el análisis, diseño y desarrollo de programas.		
Escenario:	Taller o laboratorio	Duración	6 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • 1 Computadora con procesador de texto y software de presentación para docente. • 1 Computadora cuando menos por cada 4 alumnos. • Lenguaje C++. • Compilador Visual C++. • Librerías: eclipse, OpenGL. • Poseidón community edition (software para utilizar la metodología UML). • Cañón de proyección. • Pantalla. • Escritorios. • Manuales técnicos de programas y librería. 	<ul style="list-style-type: none"> • Integra equipos de trabajo de 4 participantes: • Aplica las medidas de seguridad e higiene en el desarrollo de la práctica. • Prepara el equipo a emplear, los instrumentos, las herramientas los materiales y equipos en las mesas de trabajo. <p>Desarrollo de un programa orientado a objetos, relacionando la metodología con el lenguaje C++.</p> <p>Representa el siguiente programa con UML.</p> <div data-bbox="905 914 1841 1365" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center; background-color: #c8e6c9; margin: 0;">Programa de ejemplo</p> <pre style="margin: 0;">#include <iostream.h> /* ***** clase posición ***** */ class posicion { private: int X,Y; public: // característica de lenguaje hibrido void iniciar (int iniciarX, int iniciarY) { X = iniciarX; Y = iniciarY; }; int leerX();</pre> </div>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre> int leerY(); virtual void mensaje(){ cout << "Hola estoy en posicion\n"; }; // fin posición int posicion::leerY() { return Y; } int posicion::leerX() { return X; } /* ***** clase punto ***** */ class punto : public posicion { int visible; public: virtual void mostrar(); virtual void ocultar(); int EsVisible(); virtual void saluda(){ cout << " Hola soy un punto\n"; }; // fin punto void punto::mostrar() { visible = 1; }; void punto::ocultar() { visible = 0; }; int punto::EsVisible() { return visible; } /* ***** clase circulo ***** */ </pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre> class circulo : public punto { int radio; public: void iniciar (int iniciarX, int iniciarY, int IniciaRadio) { punto::iniciar(iniciarX, iniciarY); radio = IniciaRadio; }; void expandir(int ExpandirPor); void comprimir(int ComprimirPor); virtual void saluda(){ cout << " Hola soy un circulo\n";} }; void circulo::expandir(int ExpandirPor) { radio = radio + ExpandirPor; }; void circulo::comprimir(int ComprimirPor) { radio = radio - ComprimirPor; } /* ***** fin clase circulo ***** */ void main (void) { int Z; circulo posicionA; posicionA.iniciar(16,3,35); posicionA.ocultar(); if (posicionA.EsVisible()) { Z = posicionA.leerX(); cout << "\n valor visible de X " << Z; posicionA.saluda(); } </pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre data-bbox="905 310 1843 440" style="border: 1px solid black; padding: 5px;"> else cout << "\n valor X no visible " ; } </pre> <p data-bbox="846 509 1339 537">Compilación y ejecución del programa.</p> <ul data-bbox="800 550 1934 915" style="list-style-type: none"> • Compila y ejecuta el programa anterior, ubicando cada parte del programa bajo los siguientes aspectos, si quedara pendiente algún aspecto, proponer la parte faltante: • Programa principal. • Definición de funciones miembro. • Declaración de clases. • Archivos incluidos. • Depuración de errores • Una vez ejecutado el programa imprime la pantalla del resultado mediante la función Impr Pant de la computadora en la que esta trabajando. • Aplica la metodología de Programación Orientada a Objetos a un proyecto de aplicación real elegido por él mismo de acuerdo a su entorno. <p data-bbox="846 984 1318 1011">Elaboración del reporte de la práctica</p> <ul data-bbox="800 1032 1934 1167" style="list-style-type: none"> • Realiza el reporte de la práctica, incluyendo en éste la pantalla obtenida en el punto anterior, así como la descripción de un proyecto de aplicación real elegido por él mismo de acuerdo a su entorno. • Limpia y guarda los instrumentos de trabajo.

II. Guía de evaluación del módulo Programación básica orientada a objetos

7. Descripción

La guía de evaluación es un documento que define el proceso de recolección y valoración de las evidencias requeridas por el módulo desarrollado y tiene el propósito de guiar en la evaluación de las competencias adquiridas por los alumnos, asociadas a los Resultados de Aprendizaje; en donde además, describe las técnicas y los instrumentos a utilizar y la ponderación de cada actividad de evaluación. Los Resultados de Aprendizaje se definen tomando como referentes: las competencias genéricas que va adquiriendo el alumno para desempeñarse en los ámbitos personal y profesional que le permitan convivir de manera armónica con el medio ambiente y la sociedad; las disciplinares, esenciales para que los alumnos puedan desempeñarse eficazmente en diversos ámbitos, desarrolladas en torno a áreas del conocimiento y las profesionales que le permitan un desempeño eficiente, autónomo, flexible y responsable de su ejercicio profesional y de actividades laborales específicas, en un entorno cambiante que exige la multifuncionalidad.

La importancia de la evaluación de competencias, bajo un enfoque de **mejora continua**, reside en que es un proceso por medio del cual se obtienen y analizan las evidencias del desempeño de un alumno con base en la guía de evaluación y rúbrica, para emitir un juicio que conduzca a tomar decisiones.

La evaluación de competencias se centra en el desempeño real de los alumnos, soportado por evidencias válidas y confiables frente al referente que es la guía de evaluación, la cual, en el caso de competencias profesionales, está asociada con una norma técnica de competencia laboral (NTCL), de institución educativa o bien, una normalización específica de un sector o área y no en contenidos y/o potencialidades.

El **Modelo de Evaluación** se caracteriza porque es **Confiable** (que aplica el mismo juicio para todos los alumnos), **Integral** (involucra las dimensiones intelectual, social, afectiva, motriz y axiológica), **Participativa** (incluye autoevaluación, coevaluación y heteroevaluación), **Transparente** (congruente con los aprendizajes requeridos por la competencia), **Válida** (las evidencias deben corresponder a la guía de evaluación).

Evaluación de los Aprendizajes.

Durante el proceso de enseñanza - aprendizaje es importante considerar tres categorías de evaluación: **diagnóstica, formativa y sumativa**.

La evaluación **diagnóstica** nos permite establecer un **punto de partida** fundamentado en la detección de la situación en la que se encuentran nuestros alumnos. Permite también establecer vínculos socio-afectivos entre el docente y su grupo. El alumno a su vez podrá obtener información sobre los aspectos donde deberá hacer énfasis en su dedicación. El docente podrá **identificar las características del grupo y orientar adecuadamente sus estrategias**. En esta etapa pueden utilizarse mecanismos informales de recopilación de información.

La evaluación **formativa** se realiza durante todo el proceso de aprendizaje del alumno, en forma constante, ya sea al finalizar cada actividad de aprendizaje o en la integración de varias de éstas. Tiene como finalidad **informar a los alumnos de sus avances** con respecto a los aprendizajes que deben alcanzar y advertirle sobre dónde y en qué aspectos tiene debilidades o dificultades para poder regular sus procesos. Aquí se admiten errores, se

Identifican y se corrigen; es factible trabajar colaborativamente. Asimismo, el docente puede asumir nuevas estrategias que contribuyan a mejorar los resultados del grupo.

Finalmente, la evaluación **sumativa** es adoptada básicamente por una función social, ya que mediante ella se asume una acreditación, una promoción, un fracaso escolar, índices de deserción, etc., a través de **criterios estandarizados y bien definidos**. Las evidencias se elaboran en forma individual, puesto que se está asignando, convencionalmente, un criterio o valor. Manifiesta la síntesis de los logros obtenidos por ciclo o período escolar.

Heteroevaluación, Coevaluación y Autoevaluación

En esta nueva versión (02) de la guía de evaluación se están incluyendo de manera formal tres modalidades de evaluación, que según la persona que evalúa se denominan: heteroevaluación, coevaluación y autoevaluación.

La **heteroevaluación**: Es aquella que se realiza por personas externas al grupo escolar: representantes del sector productivo, docentes ajenos al grupo o cualquier otra persona o grupo colegiado con el dominio suficiente de la competencia, desempeño o producto que se pretenda evaluar. La heteroevaluación permite:

- Demostrar que el alumno adquirió la competencia a evaluar, en diversos contextos y ante cualquier persona o instancia evaluadora.
- Evidenciar ante agentes no integrantes del proceso enseñanza-aprendizaje las competencias desarrolladas, otorgando cierta objetividad a la evaluación.

La **coevaluación** se llevará a cabo entre pares de alumnos, pudiendo ser el evaluador un alumno o grupo de alumnos; es decir, evaluadores y evaluados intercambian su papel alternativamente. La coevaluación permite al alumno y al docente:

- Identificar los logros personales y grupales.
- Fomentar la participación, reflexión y crítica constructiva ante situaciones de aprendizaje.
- Mejorar la responsabilidad individual y de grupo.
 - Emitir juicios valorativos acerca de otros en un ambiente de libertad, compromiso y respeto.

La **autoevaluación** se refiere a la valoración que hace el alumno sobre su propia actuación o desempeño y se refiere al grado de dominio de una competencia o resultado de aprendizaje alcanzado por él mismo. Le permite al alumno:

- Reconocer sus posibilidades y limitaciones, así como definir las acciones necesarias para mejorar su aprendizaje.

Actividades de Evaluación

Los programas de estudio están conformados por Unidades de Aprendizaje (UA) que agrupan Resultados de Aprendizaje (RA) vinculados estrechamente y que requieren irse desarrollando paulatinamente. Dado que se establece un resultado, es necesario comprobar que efectivamente éste

se ha alcanzado, de tal suerte que en la descripción de cada unidad se han definido las actividades de evaluación indispensables para evaluar los aprendizajes de cada uno de los RA que conforman las unidades.

Esto no implica que no se puedan desarrollar y evaluar otras actividades planteadas por el docente, pero es importante no confundir con las actividades de aprendizaje que realiza constantemente el alumno para contribuir a que logre su aprendizaje y que, aunque se evalúen con fines formativos, no se registran formalmente en el **Sistema de Administración Escolar SAE**. El **registro formal** procede sólo para las actividades descritas en los programas y planes de evaluación.

De esta manera, los RA tienen asignada una actividad de evaluación, considerando que puede haber casos en que se incluirán dos o más RA en una sola actividad de evaluación, cuando ésta sea integradora; misma a la que se le ha determinado una ponderación con respecto a la Unidad a la cual pertenece. Ésta a su vez, tiene una ponderación que, sumada con el resto de Unidades, **conforma el 100%**. Es decir, para considerar que se ha adquirido la competencia correspondiente al módulo de que se trate, deberá **ir acumulando** dichos porcentajes a lo largo del período para estar en condiciones de acreditar el mismo. Cada una de estas ponderaciones dependerá de la relevancia que tenga la AE con respecto al RA y éste a su vez, con respecto a la Unidad de Aprendizaje. Estas ponderaciones las asignará el especialista diseñador del programa de estudios.

La ponderación que se asigna en cada una de las actividades queda asimismo establecida en la **Tabla de ponderación**, la cual está desarrollada en una hoja de cálculo que permite, tanto al alumno como al docente, ir observando y calculando los avances en términos de porcentaje, que se van alcanzando (ver apartado 7 de esta guía).

Esta tabla de ponderación contiene los Resultados de Aprendizaje y las Unidades a las cuales pertenecen. Asimismo indica, en la columna de actividades de evaluación, la codificación asignada a ésta desde el programa de estudios y que a su vez queda vinculada al Sistema de Evaluación Escolar SAE. Las columnas de aspectos a evaluar, corresponden al tipo de aprendizaje que se evalúa: **C = conceptual; P = Procedimental y A = Actitudinal**. Las siguientes tres columnas indican, en términos de porcentaje: la primera el **peso específico** asignado desde el programa de estudios para esa actividad; la segunda, **peso logrado**, es el nivel que el alumno alcanzó con base en las evidencias o desempeños demostrados; la tercera, **peso acumulado**, se refiere a la suma de los porcentajes alcanzados en las diversas actividades de evaluación y que deberá acumular a lo largo del ciclo escolar.

Otro elemento que complementa a la matriz de ponderación es la **rúbrica o matriz de valoración**, que establece los **indicadores y criterios** a considerar para evaluar, ya sea un producto, un desempeño o una actitud y la cual se explicará a continuación.

Una matriz de valoración o rúbrica es, como su nombre lo indica, una matriz de doble entrada en la cual se establecen, por un lado, los **indicadores** o aspectos específicos que se deben tomar en cuenta como **mínimo indispensable** para evaluar si se ha logrado el resultado de aprendizaje esperado y, por otro, los **criterios o niveles de calidad o satisfacción alcanzados**. En las celdas centrales se describen los criterios que se van a utilizar para evaluar esos indicadores, explicando cuáles son las características de cada uno.

Los criterios que se han establecido son: **Excelente**, en el cual, además de cumplir con los estándares o requisitos establecidos como necesarios en el logro del producto o desempeño, es propositivo, demuestra iniciativa y creatividad, o que va más allá de lo que se le solicita como mínimo, aportando elementos adicionales en pro del indicador; **Suficiente**, si cumple con los estándares o requisitos establecidos como necesarios para demostrar que se ha desempeñado adecuadamente en la actividad o elaboración del producto. Es en este nivel en el que podemos decir que se ha adquirido la competencia. **Insuficiente**, para cuando no cumple con los estándares o requisitos mínimos establecidos para el desempeño o producto.

Evaluación mediante la matriz de valoración o rúbrica

Un punto medular en esta metodología es que al alumno se le proporcione el **Plan de evaluación**, integrado por la **Tabla de ponderación y las Rúbricas**, con el fin de que pueda conocer qué se le va a solicitar y cuáles serán las características y niveles de calidad que deberá cumplir para demostrar que ha logrado los resultados de aprendizaje esperados. Asimismo, él tiene la posibilidad de autorregular su tiempo y esfuerzo para recuperar los aprendizajes no logrados.

Como se plantea en los programas de estudio, en una **sesión de clase previa a finalizar la unidad**, el docente debe hacer una **sesión de recapitulación** con sus alumnos con el propósito de valorar si se lograron los resultados esperados; con esto se pretende que el alumno tenga la oportunidad, en caso de no lograrlos, de rehacer su evidencia, realizar actividades adicionales o repetir su desempeño nuevamente, con el fin de recuperarse de inmediato y no esperar hasta que finalice el ciclo escolar acumulando deficiencias que lo pudiesen llevar a no lograr finalmente la competencia del módulo y, por ende, no aprobarlo.

La matriz de valoración o rúbrica tiene asignadas a su vez valoraciones para cada indicador a evaluar, con lo que el docente tendrá los elementos para evaluar objetivamente los productos o desempeños de sus alumnos. Dichas valoraciones están también vinculadas al SAE y a la matriz de ponderación. Cabe señalar que **el docente no tendrá que realizar operaciones matemáticas para el registro de los resultados de sus alumnos**, simplemente deberá marcar en cada celda de la rúbrica aquella que más se acerca a lo que realizó el alumno, ya sea en una hoja de cálculo que emite el SAE o bien, a través de la Web.

8. Tabla de ponderación

UNIDAD	RA	ACTIVIDAD DE EVALUACIÓN	ASPECTOS A EVALUAR			% Peso Especifico	% Peso Logrado	% Peso Acumulado
			C	P	A			
1. Manejo de componentes y estructuras básicas de programación orientada a objetos.	1.1 Identifica los fundamentos de la programación orientada a objetos, considerando la estructura de programas bajo este ambiente							
	1.2 Maneja los principios de la programación orientada a objetos	1.2.1	▲	▲	▲	20%		
% PESO PARA LA UNIDAD						20%		
2. Creación de programas orientados a objetos	2.1 Maneja los modelos de la programación orientada a objetos en el análisis, diseño y desarrollo de programas	2.1.1	▲	▲	▲	40%		
	2.2 Desarrolla programas de aplicación final, empleando los fundamentos y características de un lenguaje Orientado a Objetos para dar solución a problemas de aplicación real.	2.2.1	▲	▲	▲	40%		
% PESO PARA LA UNIDAD						60%		
PESO TOTAL DEL MÓDULO						100%		

**9. Materiales para el
desarrollo de actividades
de evaluación**

10. Matriz de valoración ó rúbrica

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	PBOO	Nombre del módulo:	Programación básica orientada a objetos.	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	1.2. Maneja los principios de la programación orientada a objetos.		Actividad de evaluación:	1.2.1 Elabora un programa básico describiendo los componentes y características de la programación orientada a objetos.	

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Identificación del entorno de los objetos y clases en Programación Orientada a Objetos	15%	Identifica en todas las figuras: objetos: estructura, organización; mensajes entre objetos; tipos de objeto, organización de los objetos. Cuida y promueve entre sus compañeros el cuidado de las instalaciones, de los materiales y del equipo.	Identifica en todas las figuras: objetos: estructura, organización; mensajes entre objetos; tipos de objeto, organización de los objetos.	No identifica en todas las figuras alguno de los siguientes aspectos: objetos: estructura, organización; mensajes entre objetos; tipos de objeto, organización de los objetos.
Identificación del entorno de los clases en Programación Orientada a Objetos	35%	Identifica en todas las figuras: Clases de: atributos, métodos, públicas, privadas, interfase, estructuras, constructores y destructores.	Identifica en todas las figuras: Clases de: atributos, métodos, públicas, privadas, interfase, estructuras, constructores y destructores.	Omite alguna de las siguientes actividades: <ul style="list-style-type: none"> Identificar en todas las figuras alguno de los siguientes aspectos: Clases: atributos,

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
		Identifica propiedades: abstracción, encapsulación, modularidad, jerarquía, herencia, polimorfismo. Ordena la información de acuerdo a categorías, jerarquías y relaciones.	Propiedades: abstracción, encapsulación, modularidad, jerarquía, herencia, polimorfismo.	métodos, públicas, privadas, interfase, estructuras, constructores y destructores. <ul style="list-style-type: none"> Identificar propiedades: abstracción, encapsulación, modularidad, jerarquía, herencia, polimorfismo.
Presentación de programa con la identificación de componentes.	40%	Describe el programa relacionando cada una de sus partes con los elementos y propiedades de los objetos y clases, así como de la programación orientada a objetos. Discrimina la información relevante de la no relevante, presentando la información organizada secuencialmente y jerárquicamente.	Describe el programa relacionando cada una de sus partes con los elementos y propiedades de los objetos y clases, así como de la programación orientada a objetos.	No describe el programa relacionando cada una de sus partes con los elementos y propiedades de los objetos y clases, así como de la programación orientada a objetos.
Coevaluación Competencias genéricas: 1. Se conoce y valora a sí mismo y aborda problemas y retos teniendo en cuenta los objetivos que persigue.	10%	Actúa con perseverancia y dedicación para lograr los objetivos que se propone. Identifica los errores, aprende de ellos y toma nuevas decisiones.	Actúa con perseverancia para lograr los objetivos que se propone. Identifica los errores y toma nuevas decisiones.	Omite alguna de las siguientes actividades: <ul style="list-style-type: none"> Actuar con perseverancia para lograr los objetivos que se propone. Identificar los errores y tomar nuevas decisiones.
	100%			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	PBOO	Nombre del módulo:	Programación básica orientada a objetos.	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	2.1 Maneja los modelos de la programación orientada a objetos en el análisis, diseño y desarrollo de programas.			Actividad de evaluación:	2.1.1 Desarrolla un programa en lenguaje C++, manejando las propiedades de objetos y clases

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Identificación elementos, instrucciones y operadores en C++.	25%	<p>Describe la sintaxis y forma en que operan las instrucciones y desarrolla 5 ejemplos de aplicación de las mismas</p> <p>Describe línea por línea lo que realiza el programa representado, y el resultado que entrega.</p> <p>Identifica la representación de clases en el programa.</p> <p>Describe elementos, objetos y los resultados que entrega el programa.</p> <p>Deduca datos técnicos o procedimientos, tanto en su idioma natal como en idioma inglés.</p>	<p>Describe la sintaxis y forma en que operan las instrucciones y desarrolla 5 ejemplos de aplicación de las mismas</p> <p>Describe línea por línea lo que realiza el programa representado, y el resultado que entrega.</p> <p>Identifica la representación de clases en el programa.</p> <p>Describe elementos, objetos y los resultados que entrega el programa.</p>	<p>Omite alguno de los siguientes aspectos:</p> <ul style="list-style-type: none"> • Describir la sintaxis y forma en que operan las instrucciones y desarrolla 5 ejemplos de aplicación de la misma. • Describir línea por línea lo que realiza el programa representado, y el resultado que entrega. • Identificar la representación de clases en el programa. • Describir elementos, objetos y los resultados que entrega el programa.
Identificación de la aplicación de la herencia y	35%	Identifica cada elemento del programa, la función que realiza en	Identifica cada elemento del programa, la función que realiza en	Omite alguno de los siguientes aspectos:

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
el polimorfismo		<p>él y los resultados que entrega.</p> <p>Describe todos los elementos, componentes, fundamentos y propiedades de la programación orientada a objetos.</p> <p>Demuestra el conocimiento y dominio de palabras, nemotécnicos y silogismos propios del ámbito profesional en el que se está formando.</p>	<p>él y los resultados que entrega.</p> <p>Describe todos los elementos, componentes, fundamentos y propiedades de la programación orientada a objetos.</p>	<ul style="list-style-type: none"> Identificar cada elemento del programa, la función que realiza en él y los resultados que entrega. Describir todos los elementos, componentes, fundamentos y propiedades de la programación orientada a objetos.
Ejecución compilador del Lenguaje C++	20%	<p>Ejecuta programa fuente, obtiene programa objeto, ejecuta el programa e imprime los resultados para los programas de esta práctica.</p> <p>Reporta resultados del manejo de los conceptos y resultados de ejecución del programa.</p> <p>Sigue las instrucciones y el procedimiento de edición de programas en Lenguaje C++ de manera sistemática, evitando acciones no consideradas en los mismos que pudieran provocar errores en la compilación.</p>	<p>Ejecuta programa fuente, obtiene programa objeto, ejecuta el programa e imprime los resultados para los programas de esta práctica.</p> <p>Reporta resultados del manejo de los conceptos y resultados de ejecución del programa.</p>	<p>Omite alguno de los siguientes aspectos:</p> <ul style="list-style-type: none"> Ejecutar programa fuente. Obtener programa objeto. Ejecutar el programa. Imprimir los resultados para los programas de esta práctica. Reportar resultados del manejo de los conceptos y resultados de ejecución del programa.
Manejo de la metodología UML de representación de modelos, en el modelaje y desarrollo de programas	20%	<p>Realiza la identificación de los elementos de la metodología UML de representación de modelos, en el modelaje y desarrollo de programas, interpretando sus</p>	<p>Realiza la identificación de los elementos de la metodología UML de representación de modelos, en el modelaje y desarrollo de programas, interpretando sus</p>	<p>No realiza la identificación de alguno de los elementos de la metodología UML de representación de modelos, en el modelaje y desarrollo de programas, interpretando sus</p>

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
		representaciones gráficas y usos. Sintetiza evidencias obtenidas durante la identificación de los elementos de la metodología UML, emitiendo conclusiones y formulando preguntas que le permitan aclarar posibles dudas sobre el modelaje y desarrollo de programas.	representaciones gráficas y usos.	representaciones gráficas y usos.
	100%			

Siglema:	PBOO	Nombre del módulo:	Programación básica orientada a objetos.	Nombre del alumno:	
docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	2.2 Desarrolla programas de aplicación final, empleando los fundamentos y características de un lenguaje Orientado a Objetos para dar solución a problemas de aplicación real.			Actividad de evaluación:	2.2.1 Desarrolla un programa de aplicación final, empleando la metodología y técnica de un Lenguaje de programación orientada a objetos.

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Desarrollo de un programa orientado a objetos, relacionando la metodología con el lenguaje C++.	30%	<p>Representa el programa presentado con UML.</p> <p>Realiza la representación de Clases de bloques en el programa según corresponda: Elementos, Relaciones y Diagramas.</p> <p>Introduce cambios o aporta ideas para mejorar los procesos de la programa orientado a objetos, relacionando la metodología con el Lenguaje C++</p>	<p>Representa el programa presentado con UML.</p> <p>Realiza la representación de Clases de bloques en el programa según corresponda: Elementos, Relaciones y Diagramas.</p>	<p>Omite alguno de los siguientes aspectos:</p> <ul style="list-style-type: none"> • Representar el programa presentado con UML. • Realizar la representación de Clases de bloques en el programa según corresponda: Elementos, Relaciones y Diagramas.
Compilación y ejecución del programa.	40%	<p>Compila y ejecuta un programa, ubicando cada parte del mismo: Programa principal, definición de funciones miembro, declaración de clases, archivos incluidos, depuración de errores.</p> <p>Una vez ejecutado el programa</p>	<p>Compila y ejecuta un programa, ubicando cada parte del mismo: Programa principal, definición de funciones miembro, declaración de clases, archivos incluidos, depuración de errores.</p> <p>Una vez ejecutado el programa</p>	<p>Omite alguno de los siguientes aspectos:</p> <ul style="list-style-type: none"> • Compilar y ejecutar un programa, ubicando cada parte del mismo: Programa principal, definición de funciones miembro,

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
		<p>imprime la pantalla del resultado mediante la función Impr Pant de la computadora en la que está trabajando.</p> <p>Aplica la metodología de Programación Orientada a Objetos a un proyecto de aplicación real elegido por él mismo de acuerdo a su entorno.</p> <p>Asume una actitud constructiva, congruente con los conocimientos y habilidades con los que cuenta y realizando su trabajo con limpieza y pulcritud al desarrollar un proyecto de aplicación real y presentarlo como trabajo final del curso.</p>	<p>imprime la pantalla del resultado mediante la función Impr Pant de la computadora en la que está trabajando.</p> <p>Aplica la metodología de Programación Orientada a Objetos a un proyecto de aplicación real elegido por él mismo de acuerdo a su entorno.</p> <p>Asume una actitud constructiva, congruente con los conocimientos y habilidades con los que cuenta y realizando su trabajo con limpieza y pulcritud al desarrollar un proyecto de aplicación real y presentarlo como trabajo final del curso.</p>	<p>declaración de clases, archivos incluidos, depuración de errores.</p> <ul style="list-style-type: none"> • Imprimir la pantalla del resultado mediante la función Impr Pant de la computadora en la que está trabajando. • Aplicar la metodología de Programación Orientada a Objetos a un proyecto de aplicación real elegido por él mismo de acuerdo a su entorno.
Elaboración del reporte de la práctica.	30%	<p>Realiza el reporte de la práctica, incluyendo las pantallas obtenidas, así como la descripción de un proyecto de aplicación real elegido por él de acuerdo a su entorno.</p> <p>Limpia y guarda los instrumentos de trabajo.</p> <p>Estructura ideas y argumentos de manera clara, coherente y sintética al elaborar su reporte.</p>	<p>Realiza el reporte de la práctica, incluyendo las pantallas obtenidas, así como la descripción de un proyecto de aplicación real elegido por él de acuerdo a su entorno.</p> <p>Limpia y guarda los instrumentos de trabajo.</p>	<p>Omite alguno de los siguientes aspectos:</p> <ul style="list-style-type: none"> • Realizar el reporte de la práctica, incluyendo las pantallas obtenidas, así como la descripción de un proyecto de aplicación real elegido por él de acuerdo a su entorno. • Limpiar y guardar los instrumentos de trabajo.
	100%			