



GOBIERNO DE
MÉXICO

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



Guía pedagógica y de evaluación del módulo

Programación orientada a objetos

Currículum Laboral

Áreas:

Tecnología y transporte

Carrera:

Profesional Técnico-Bachiller en
Informática

4º semestre

Editor: Colegio Nacional de Educación Profesional Técnica

Módulo: Programación orientada a objetos.

Área: Tecnología y transporte.

Carrera: PT-B en Informática

Semestre: 4°

Horas por semana: 8

Fecha de diseño o actualización: 31 de mayo de 2024.

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

© Colegio Nacional de Educación Profesional Técnica

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

Directorio

Arturo Pontifes Martínez
Dirección General

Camilo García Ramírez
Secretaría General

Hugo Nicolás Pérez González
Secretaría Académica

Patricia Alejandra Bernal Monzón
Dirección de Diseño Curricular

Programación orientada a objetos.

Contenido

	Pág.
I Guía pedagógica	
1 Descripción	5
2 Generalidades pedagógicas	6
3 Orientaciones didácticas	8
4 Estrategias de aprendizaje	10
5 Prácticas y Actividades	13
II Guía de evaluación	
6 Descripción	14
7 Tabla de ponderación	16
8 Matriz de valoración o rúbrica	17

I. Guía pedagógica

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 y el proceso de aprendizaje en el alumnado en el desarrollo de habilidades previstas en los programas de estudio.

Tomando en consideración el Marco Curricular Común de la Educación Media Superior (MCCEMS) el docente asume el rol de diseñador didáctico, innovador educativo, agente de transformación social, el cual se rige por principios orientadores, acompañando al estudiantado hacia una participación activa que potencialice su desarrollo; identificando los intereses y necesidades de aprendizaje que le lleven a resolver desafíos en su contexto, favoreciendo con ello el modelo de una escuela abierta, que atienda a la diversidad cultural, lingüística, de género, a la interacción entre grupos sociales, la coherencia entre los valores y objetivos de cada módulo.

Considerando al alumnado como protagonista para la transformación social, a través del desarrollo de un pensamiento crítico, analítico y flexible, se busca acercarle elementos de apoyo que le muestren cómo desarrollar **habilidades, conocimientos, actitudes y valores** en un contexto específico. Mediante la guía pedagógica el alumno podrá **autogestionar su aprendizaje** por medio del uso de estrategias flexibles y apropiadas que se puedan transferir y adoptar a nuevas situaciones y contextos, e ir dando seguimiento a sus avances a través de la autoevaluación, la coevaluación y la evaluación formativa.

2. Generalidades pedagógicas

Nuestro modelo académico sustenta una base pedagógica centrada en la teoría constructivista con un enfoque humanista, la cual tiene presente la diversidad local, regional, nacional e internacional, combinada con el nuevo MCCEMS permitirá mantener una didáctica que apuesta por el desarrollo de la voluntad de aprender, hacer una conexión entre el contenido teórico y la realidad.

Se pretende fomentar un aprendizaje, situado, profundo y significativo, que conlleve a la transversalidad incitando al desarrollo de estrategias de enseñanza basadas en proyectos integradores, que articulen los conocimientos con las unidades de aprendizaje y con los recursos socioemocionales, que orienten a la formación integral del estudiantado.

El alumnado adquiere un rol protagónico del proceso educativo, guiándolo al involucramiento y resolución de problemas económicos, políticos, sociales y del medio ambiente para la construcción de un mundo más justo, pacífico y sostenible, bajo el acompañamiento, orientación y conducción del docente, por lo que el docente bajo su experiencia buscará una combinación de estrategias didácticas que incorporen materiales y recursos que den sentido a lo que el estudiante aprende.

De acuerdo con lo anterior, se debe considerar que el papel que juega el alumnado y el personal docente en el marco del Modelo Académico del CONALEP tenga, entre otras, las siguientes características:

El alumnado:

- ❖ Gestiona su aprendizaje permanente.
- ❖ Mejora su capacidad para resolver problemas.
- ❖ Trabaja de forma colaborativa.
- ❖ Se comunica asertivamente.
- ❖ Busca información actualizada de fuentes confiables.
- ❖ Construye su conocimiento.
- ❖ Adopta una posición crítica, autónoma y propositiva.
- ❖ Realiza responsablemente los procesos de autoevaluación y coevaluación.
- ❖ Se vuelve agente de transformación social.
- ❖ Actúa con valores y principios éticos.
- ❖ Practica hábitos saludables para el autocuidado.
- ❖ Construye un pensamiento crítico, analítico y flexible.

El personal docente:

- ❖ Considera necesidades e intereses de los estudiantes que propicien la motivación y participación.
- ❖ Domina y estructura los saberes para facilitar experiencias de aprendizaje.
- ❖ Planifica los procesos de enseñanza dirigidos al logro de resultados de aprendizaje de manera efectiva, creativa e innovadora aplicado a su contexto.
- ❖ Evalúa los aprendizajes con un enfoque formativo, retroalimentando para la búsqueda de la mejora continua.
- ❖ 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.
- ❖ Propone proyectos integradores en búsqueda de la transversalidad, para la solución de problemáticas contextuales, vinculadas a la comunidad generando el sentido de la experimentación pedagógica.
- ❖ Utiliza tecnologías de la información y comunicación, tecnologías de aprendizaje y conocimiento, tecnologías del empoderamiento y participación, como recursos didácticos.
- ❖ Agente de transformación social.
- ❖ Participa de forma colaborativa en el trabajo de academias.

3. Orientaciones didácticas

Para el logro del propósito de cada **unidad de aprendizaje** del módulo, se recomienda al personal docente lo siguiente:

- Identificar los componentes básicos de los resultados de aprendizaje para realizar el plan clase, considerando los elementos con los que se puede trabajar el contenido.
- Abordar conocimientos previos a través del diseño de una actividad, considerando la exploración de conocimientos, saberes e ideas precedentes.
- Seleccionar actividades pertinentes y acordes a los resultados de aprendizaje, procurando activar la atención del estudiantado a partir de generar ambientes de trabajo encaminados a la reflexión, el diálogo y la discusión.
- Considerar métodos y estrategias que favorezcan aprendizajes significativos.
- Plantear el objetivo de cada actividad buscando la contextualización de acuerdo con las características de la comunidad, municipio, región y estados.
- Plantear actividades dirigidas al trabajo directo con la comunidad de forma independiente como un complemento a lo revisado en clase o una extensión del tema; de ser posible tener un repositorio de información digital para alojar los materiales que el estudiantado deba consultar.
- Retroalimentar las actividades y trabajos del estudiantado con el fin de orientarlos sobre sus avances y aspectos a mejorar en sus procesos de aprendizaje.
- Promover la coevaluación, autoevaluación y heteroevaluación para favorecer la retroalimentación formativa y asertiva.
- Aplicar la transversalidad buscando proyectos que se interrelacionen de forma horizontal y vertical basado en el mapa curricular.
- Procurar que las actividades realizadas de forma independiente sean un complemento a lo revisado en clase o una extensión del tema y deberán estar dirigidas al trabajo directo con la comunidad.
- Compartir los propósitos educativos y los criterios del logro de aprendizaje con los estudiantes.
- Diseñar e implementar actividades que evidencien lo que el estudiantado está aprendiendo.
- Procurar incluir el aprendizaje práctico fuera del aula, intercambiar conocimientos con miembros de la comunidad, generar dinámicas con amigos, vecinos o familiares, ejecutar actividades comunicativas y académicas específicas, así como la aplicación progresiva de

métodos didácticos; es importante observar e identificar las habilidades y aptitudes de los estudiantes para encaminarlos, desarrollarlas mejor y apoyarles.

- Algunas estrategias para la utilización de la retroalimentación formativa son las siguientes:
 1. Clarificar y compartir los objetivos de aprendizaje y criterios de desempeño con cada estudiante al inicio de cada resultado de aprendizaje.
 2. Diseñar discusiones de clase efectivas, preguntas, actividades y tareas que hagan evidente el aprendizaje del estudiante.
 3. Proveer retroalimentación que motive el aprendizaje.
 4. Activar en la comunidad estudiantil el deseo de ser responsables de su propio proceso de aprendizaje.
 5. Fomentar la participación de las y los estudiantes como recurso de apoyo para sus pares.
- Conforme a los preceptos del MCCEMS, retomar los Recursos Socioemocionales que conforman el currículum ampliado: la Responsabilidad Social, el Cuidado Físico Corporal y el Bienestar Emocional Afectivo, constituyendo un eje articulador el cual busca que las y los estudiantes se formen como ciudadanas y ciudadanos responsables, honestos, comprometidos con el bienestar físico mental y emocional, tanto personal como social. Se pretende trabajar con mayor autonomía en el aula, la escuela, la comunidad, la solidaridad, la inclusión y la diversidad, así como el reconocimiento de la perspectiva de género y los aportes de la cultura de paz, de valorar el esfuerzo de las conductas legales y del trabajo justo y honrado, al poner en práctica acciones ciudadanas y proyectos escolares comunitarios.
- Derivado de lo anterior, durante el desarrollo del módulo se sugiere tener presente el Currículum ampliado, establecido en el Acuerdo número 09/08/23 por el que se establece y regula el Marco Curricular Común de la Educación Media Superior https://www.dof.gob.mx/nota_detalle.php?codigo=5699835&fecha=25/08/2023#gsc.tab=0

4. Estrategias de aprendizaje

Para el desarrollo del resultado de aprendizaje 1.1, se recomienda al alumnado:

- Realizar una investigación documental sobre la metodología de la programación orientada a objetos y los elementos empleados en el desarrollo de aplicaciones, describiendo los siguientes elementos:
 - Programación orientada a Objetos
 - Objetos
 - Mensajes
 - Paso de mensajes
 - Estructura interna de los objetos
 - Herencia
 - Polimorfismo
 - Clases
 - Diferencia entre clases y objetos
 - Lenguaje de Modelado
- Realizar el modelado de diagrama de casos de uso de una tienda de electrodomésticos.
- Desarrollar el modelado de diagrama de casos de uso de una cadena de videoclubes.
- Esquematizar el modelado de diagrama de casos de uso de una máquina expendedora y de venta de bebidas.
- Diseñar el modelado de diagrama de clases del evento de la inscripción escolar.
- Hacer el modelado de diagrama de clases de un estacionamiento público.
- Elaborar el modelado de diagrama de secuencia de un centro de instalaciones deportivas.
- Crear el modelado de diagrama de secuencia de un inicio de sesión en un sistema.
- Investigar al menos 3 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos en el que describa:
 - Plataforma
 - Editor de texto empleado
 - Compilador
 - Enlazador
 - Entornos de Desarrollo Integrado
- **Realizar la actividad de evaluación 1.1.1 considerando la rúbrica correspondiente**

Para el desarrollo del resultado de aprendizaje 1.2, se recomienda al alumnado:

- Aplicar en un modelo las siguientes estructuras manejadas en el lenguaje de programación seleccionado:
 - Funciones de entrada y salida estándar.
 - Estructuras de decisión para definir el orden de ejecución de bloques de código:
 - if – else
 - else – if
 - switch
 - Estructuras de ciclo para la ejecución continua y/o repetida de bloques de código:
 - while
 - do – while
 - for
 - Ciclos infinitos
 - Tipos de datos estructurados:
 - Arreglo
 - Estructuras
 - Clases basado en POO
 - Aplicaciones implementando los componentes de la programación orientada a objetos, donde aplica:
 - Funciones de entrada y salida
 - Sentencias de control
 - Arreglos y estructuras
 - Clases y objetos
 - Sentencias en el manejo de errores de excepción:
 - class
 - private
 - public
 - protected
- Realizar la actividad de evaluación 1.2.1 considerando la rúbrica correspondiente

Para el desarrollo del resultado de aprendizaje 2.1, se recomienda al alumnado:

- Aplicar en un modelo los siguientes elementos de la programación orientada a objetos en el lenguaje de programación seleccionado:
 - Manejo de herencia basado en POO
 - Sobrecarga de operadores
 - Sobrecarga de operadores + y - basado en POO
 - Sobrecarga de operadores +, ==, ++ y << basado en POO
 - Suma y resta de números complejos basado en POO
 - Polimorfismo realizando cálculo de volumen de figuras geométricas basado en POO
 - Crear una clase abstracta empleando polimorfismo basado en POO
 - Desarrollar programas de cómputo utilizando las propiedades de herencia y polimorfismo
- **Realizar la actividad de evaluación 2.1.1 considerando la rúbrica correspondiente.**

Para el desarrollo del resultado de aprendizaje 2.2, se recomienda al alumnado:

- Investigar los siguientes conceptos y elementos de seguridad aplicados en la implementación de programas orientados a objetos utilizando el lenguaje de programación seleccionado:
 - Patrones de diseño más comunes empleados para la autenticación de aplicaciones orientadas a objetos:
 - Fábricas abstractas
 - Singleton o Singular
 - Factory Method
 - Prototipo
 - Algoritmos de cifrado de datos y lo implementa en alguna aplicación.
- Aplicar en un modelo las siguientes sentencias en el manejo de errores de excepción:
 - try
 - catch
 - throw
- Ejecutar en un programa los siguientes aspectos:
 - Manejo de errores de excepción basado en POO
 - Manejo de excepciones en clases basado en POO
 - Medidas de seguridad en los programas diseñados
- **Realizar la actividad de evaluación 2.2.1 considerando la rúbrica correspondiente**

5. Prácticas y Actividades

En respeto a la autonomía didáctica, este apartado quedará bajo la responsabilidad del personal docente para que, de acuerdo con su experiencia, características del grupo, la comunidad y el desempeño del estudiantado, seleccione, proponga y realice aquellas prácticas y actividades transversales que garanticen un mayor desarrollo de aprendizajes y habilidades, privilegiando las corrientes filosóficas, pedagógicas y técnicas de mayor actualidad, así como las tecnologías de la información y la comunicación, como herramientas de apoyo al proceso de enseñanza – aprendizaje.

Por lo anterior, se reconoce que la función del personal docente implica, ante todo, una labor de investigación y promoción del autoaprendizaje; fomentando actividades que consideren el aprendizaje contextualizado, colaborativo, participativo y lúdico, así como el diálogo, el trabajo en equipo y la utilización pertinente, sostenible y responsable de las tecnologías de la información y comunicación, conocimiento y aprendizaje digital, en los procesos de la vida cotidiana con una perspectiva crítica de los contenidos y materiales disponibles en medios electrónicos, plataformas virtuales y redes sociales.

De igual manera, se espera que el estudiantado asuma su responsabilidad y tome un papel activo en el proceso de desarrollo de **habilidades, conocimientos, actitudes y valores** que le permitirán no sólo ingresar al mundo laboral, sino participar de manera destacada en la sociedad.

II. Guía de Evaluación

6. 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 orientar en la evaluación de las habilidades, conocimientos y actitudes adquiridos por el estudiantado, asociados a los Resultados de Aprendizaje; en donde, además, se describen las técnicas y los instrumentos a utilizar, así como la ponderación de cada actividad de evaluación.

Durante el proceso de enseñanza - aprendizaje es importante considerar tres finalidades 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 estudiantes. Permite también establecer vínculos socioafectivos entre el docente y su grupo. El estudiantado a su vez podrá obtener información sobre los aspectos donde deberá hacer énfasis en su dedicación. El docente podrá identificar intereses, necesidades y características del grupo para 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 estudiantado, de manera constante, ya sea al finalizar cada actividad de aprendizaje o en la integración de varias de éstas. Tiene como finalidad informar al estudiantado 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 personal docente puede asumir nuevas estrategias que contribuyan a mejorar los resultados del grupo, entendiendo que la evaluación es un proceso que construye para retroalimentar y tomar decisiones orientadas a la mejora continua, en distintos rubros.

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 claramente 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.

Con respecto al agente o responsable de llevar a cabo la evaluación, se distinguen tres categorías: la **autoevaluación** que se refiere a la valoración que hace el alumno sobre su propia actuación, lo que le permite reconocer sus posibilidades, limitaciones y cambios necesarios para mejorar su aprendizaje. Los roles de evaluador y evaluado coinciden en la misma persona.

La **coevaluación** es aquella en la que las y los alumnos se evalúan mutuamente, es decir, evaluadores y evaluados intercambian su papel alternativamente; las y los alumnos en conjunto, participan en la valoración de los aprendizajes logrados, ya sea por algunos de sus miembros o del grupo en su conjunto; la coevaluación permite al alumnado y al profesorado:

- Identificar los logros personales y grupales
- Fomentar la participación, reflexión y crítica constructiva ante situaciones de aprendizaje
- Opinar sobre su actuación dentro del grupo
- Desarrollar actitudes que promuevan la integración del grupo
- Mejorar su responsabilidad e identificación con el trabajo
- Emitir juicios valorativos acerca de otros en un ambiente de libertad, compromiso y responsabilidad

La **heteroevaluación** es el tipo de evaluación que con mayor frecuencia se utiliza, donde el docente es quien evalúa, su variante externa, se da cuando agentes no integrantes del proceso enseñanza-aprendizaje son los evaluadores, otorgando cierta objetividad por su no implicación.

En dos rúbricas diferentes de la guía de evaluación se establece un indicador específico para la autoevaluación y coevaluación; a su vez, la heteroevaluación queda establecida en una rúbrica que podría ser evaluada por un experto o docente que no haya impartido el módulo a ese grupo.

Cada uno de los Resultados de Aprendizaje (RA) tiene asignada al menos una actividad de evaluación (AE), a la que se le ha determinado una ponderación con respecto a su complejidad y relevancia. Las ponderaciones de las AE deberán sumar 100%.

7. Tabla de ponderación

La ponderación que se asigna en cada una de las actividades de evaluación se representa en la Tabla de ponderación que, además, contiene los Resultados y Unidades de aprendizaje a las cuales pertenecen. La columna “Actividad de evaluación” indica la codificación asignada a ésta desde el programa de estudios y que a su vez queda vinculada al Sistema de Evaluación Escolar (SAE). Asimismo, la columna “Peso específico”, señala el porcentaje definido para cada actividad; la columna “Peso logrado” es el nivel que la o el alumno alcanzó con base en las evidencias o desempeños demostrados; y la columna “Peso acumulado” se refiere a la suma de los porcentajes alcanzados en las diversas actividades de evaluación a lo largo del ciclo escolar.

UNIDAD	RESULTADO DE APRENDIZAJE	ACTIVIDAD DE EVALUACIÓN	% Peso Específico	% Peso Logrado	% Peso Acumulado
1. Manejo de componentes del lenguaje orientados a objetos.	1.1 Identifica los componentes fundamentales de la programación orientada a objetos.	1.1.1	25		
	1.2 Modela y codifica programas de cómputo haciendo uso del paradigma orientado a objetos, a través del uso de sentencias de control, objetos y clases.	1.2.1	25		
% PESO PARA LA UNIDAD			50		
2. Desarrollo de aplicaciones orientadas a objetos.	2.1 Elabora aplicaciones mediante la interacción de los objetos y actores del sistema.	2.1.1	35		
	2.2 Asegura la integridad de los datos implementando medidas de seguridad haciendo uso de patrones de diseño.	2.2.2	15		
% PESO PARA LA UNIDAD			50		
PESO TOTAL DEL MÓDULO			100%		

8. Matriz de valoración o rúbrica

Otro elemento que complementa a la Tabla de ponderación es la rúbrica o matriz de valoración, que establece los indicadores y criterios a considerar para evaluar una habilidad, destreza o actitud. 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 columnas 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**, ha alcanzado el resultado de aprendizaje, además de cumplir con los estándares o requisitos establecidos como necesarios en el logro de la habilidad, destreza o actitud, es decir, va más allá de lo que se solicita como mínimo, aportando elementos adicionales en pro del indicador.
- ✓ **Bueno**, ha alcanzado el resultado de aprendizaje, es decir, cumple con los estándares o requisitos establecidos como necesarios para demostrar el logro de la habilidad, destreza o actitud.
- ✓ **Suficiente**, ha alcanzado el resultado de aprendizaje con áreas de mejora.
- ✓ **Insuficiente**, no ha logrado alcanzar el resultado de aprendizaje.

Siglema:	PROO-20	Nombre del módulo:	Programación orientada a objetos	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	1.1 Identifica los componentes fundamentales de la programación orientada a objetos.		Actividad de evaluación:	1.1.1 Determina la importancia del paradigma de la programación orientada a objetos a través del conocimiento de los principales componentes de la POO y la importancia de su modelado.	

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
Elementos de la programación orientada a objetos	30	<ul style="list-style-type: none"> Investiga minuciosamente acerca de la programación orientada a objetos y sus elementos, en la que describe la información más importante de: <ul style="list-style-type: none"> Programación orientada a objetos. Objetos. Clases. Modelo de clases. Abstracción. Encapsulación. Herencia. Polimorfismo. Relación entre clases y objetos. Elabora detalladamente un mapa conceptual con la información más importante, donde describe la forma en que interactúan los diferentes elementos de la 	<ul style="list-style-type: none"> Investiga acerca de la programación orientada a objetos y sus elementos, en la que describe la información clave de: <ul style="list-style-type: none"> Programación orientada a objetos. Objetos. Clases. Modelo de clases. Abstracción. Encapsulación. Herencia. Polimorfismo. Relación entre clases y objetos. Elabora un mapa conceptual con la información clave, donde describe la forma en que interactúan los diferentes elementos de la programación orientada a objetos. 	<ul style="list-style-type: none"> Investiga algunos elementos relativos a la programación orientada a objetos y sus elementos, en la que describe la información básica de: <ul style="list-style-type: none"> Programación orientada a objetos. Objetos. Clases. Modelo de clases. Abstracción. Encapsulación. Herencia. Polimorfismo. Relación entre clases y objetos. Elabora un mapa conceptual con la información básica, donde describe la forma en que interactúan los diferentes elementos de 	<ul style="list-style-type: none"> Omite algunos elementos en su investigación sobre la programación orientada a objetos y sus elementos, en la que describe la información mínima requerida de: <ul style="list-style-type: none"> Programación orientada a objetos. Objetos. Clases. Modelo de clases. Abstracción. Encapsulación. Herencia. Polimorfismo. Relación entre clases y objetos. Elabora un organizador gráfico diferente a un mapa conceptual con la información mínima, donde describe la forma en que interactúan los

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
		programación orientada a objetos.		la programación orientada a objetos.	diferentes elementos de la programación orientada a objetos.
Lenguaje modelado	30	<ul style="list-style-type: none"> Diseña detalladamente modelos orientados a objetos, haciendo uso de los siguientes tipos de diagramas: <ul style="list-style-type: none"> - Cajas de uso - Clases - Actividad. Presenta ordenadamente los diagramas elaborados. Emplea sistemáticamente una herramienta de modelado UML para diseñar detalladamente los diagramas y los presenta en medio magnético. 	<ul style="list-style-type: none"> Diseña modelos orientados a objetos, haciendo uso de los siguientes tipos de diagramas: <ul style="list-style-type: none"> - Cajas de uso - Clases - Actividad. Presenta los diagramas elaborados. Emplea una herramienta de modelado UML para diseñar los diagramas y los presenta en medio magnético. 	<ul style="list-style-type: none"> Elabora algunos modelos orientados a objetos, omitiendo el uso de uno de los siguientes tipos de diagramas: <ul style="list-style-type: none"> - Cajas de uso - Clases - Actividad. Presenta con errores los diagramas elaborados. Requiere apoyo para emplear una herramienta de modelado UML y diseñar los diagramas. 	<ul style="list-style-type: none"> Requiere apoyo para elaborar los modelos orientados a objetos y omite el uso de dos de los siguientes tipos de diagramas: <ul style="list-style-type: none"> - Cajas de uso - Clases - Actividad. Omite la presentación de los diagramas. Excluye el uso de una herramienta de modelado UML para diseñar los diagramas.
Software de desarrollo	30	<ul style="list-style-type: none"> Elabora jerárquicamente una tabla para describir minuciosamente al menos 3 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos contemplando los siguientes elementos: <ul style="list-style-type: none"> - Plataforma. - Editor de texto empleado. - Compilador. - Enlazador. 	<ul style="list-style-type: none"> Elabora una tabla para describir con información clave, al menos 3 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos contemplando los siguientes elementos: <ul style="list-style-type: none"> - Plataforma. - Editor de texto empleado. - Compilador. - Enlazador. 	<ul style="list-style-type: none"> Elabora una tabla para describir con información básica, 2 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos contemplando los siguientes elementos: <ul style="list-style-type: none"> - Plataforma. - Editor de texto empleado. - Compilador. - Enlazador. 	<ul style="list-style-type: none"> Elabora una tabla para describir con la información mínima requerida, 2 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos, omitiendo alguno de los siguientes elementos: <ul style="list-style-type: none"> - Plataforma. - Editor de texto empleado. - Compilador. - Enlazador.

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
		<ul style="list-style-type: none"> - Entornos de desarrollo integrados. • Presenta ordenadamente la tabla elaborada. • Relata un análisis detallado de las ventajas de portabilidad entre las diferentes plataformas que presentan algunos lenguajes de programación orientados a objetos. 	<ul style="list-style-type: none"> - Entornos de desarrollo integrados. • Presenta la tabla elaborada. • Relata un análisis de las ventajas de portabilidad entre las diferentes plataformas que presentan algunos lenguajes de programación orientados a objetos. 	<ul style="list-style-type: none"> - Entornos de desarrollo integrados. • Presenta la tabla elaborada con errores. • Relata un resumen de algunas de las ventajas de portabilidad entre las diferentes plataformas que presentan algunos lenguajes de programación orientados a objetos. 	<ul style="list-style-type: none"> - Entornos de desarrollo integrados. • Omite la presentación de la tabla elaborada. • Relata con errores un resumen con la información básica de algunas ventajas de portabilidad entre las diferentes plataformas que presentan algunos lenguajes de programación orientados a objetos.
Presentación de resultados (Autoevaluación)	10	<ul style="list-style-type: none"> • Presenta detalladamente la información más importante descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso y digital. • Elabora minuciosamente el documento con las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<ul style="list-style-type: none"> • Presenta la información clave descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso y digital. • Elabora el documento con las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<ul style="list-style-type: none"> • Presenta la información básica descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso o digital. • Elabora el documento omitiendo alguna de las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<ul style="list-style-type: none"> • Presenta la información mínima requerida descrita en los apartados anteriores en un formato diferente a un organizador gráfico, elaborado en un procesador de texto, en formato impreso o digital. • Elabora el documento con errores, omitiendo dos o más de las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente electrónicas empleadas.
	100				

Siglema:	PROO-20	Nombre del módulo:	Programación orientada a objetos	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	1.2 Modela y codifica programas de cómputo haciendo uso del paradigma orientado a objetos, a través del uso de sentencias de control, objetos y clases.			Actividad de evaluación:	1.2.1 Desarrolla aplicaciones implementando los componentes de la programación orientada a objetos donde aplica: funciones de entrada y salida, sentencias de control, manejo de arreglos y estructuras, clases y objetos.

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
Funciones de entrada y salida	15	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla minuciosamente programas con funciones de entrada y salida. – Verifica detalladamente que el código se ejecuta correctamente y en caso de errores corrige. – Presenta ordenadamente el código fuente en sintaxis correcta. • Inserta comentarios detallados en el código desarrollado para facilitar 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas con funciones de entrada y salida. – Verifica que el código se ejecuta correctamente y en caso de errores corrige. – Presenta el código fuente en sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas con funciones de entrada y salida. – Verifica que el código se ejecuta. – Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas con funciones de entrada y salida. – Identifica la ejecución del código. – Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
		la comprensión y la documentación de la programación realizada.			
Sentencias de control	15	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla minuciosamente programas utilizando sentencias de control: <ul style="list-style-type: none"> ○ If else ○ else if ○ switch ○ for ○ while ○ do while – Verifica detalladamente que el código se ejecuta correctamente, y en caso de errores corrige. – Presenta ordenadamente el código fuente y la sintaxis correcta de la aplicación desarrollada. • Inserta comentarios detallados en el código 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas utilizando sentencias de control: <ul style="list-style-type: none"> ○ If else ○ else if ○ switch ○ for ○ while ○ do while – Verifica que el código se ejecuta correctamente, y en caso de errores corrige. – Presenta el código fuente y la sintaxis correcta de la aplicación desarrollada. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas utilizando sentencias de control: <ul style="list-style-type: none"> ○ If else ○ else if ○ switch ○ for ○ while ○ do while – Verifica que el código se ejecuta. – Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas utilizando sentencias de control: <ul style="list-style-type: none"> ○ If else ○ else if ○ switch ○ for ○ while ○ do while – Identifica la ejecución del código. – Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
		desarrollado para facilitar la comprensión y la documentación de la programación realizada.			
Arreglos y estructuras	15	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla jerárquicamente programas que hacen uso de: <ul style="list-style-type: none"> ○ Arreglos ○ Estructuras – Verifica detalladamente que el código se ejecuta correctamente, y en caso de errores corrige. – Presenta ordenadamente el código fuente y la sintaxis correcta de la aplicación desarrollada. • Inserta comentarios detallados en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas que hacen uso de: <ul style="list-style-type: none"> ○ Arreglos ○ Estructuras – Verifica que el código se ejecuta correctamente, y en caso de errores corrige. – Presenta el código fuente y la sintaxis correcta de la aplicación desarrollada. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas que hacen uso de: <ul style="list-style-type: none"> ○ Arreglos ○ Estructuras – Verifica que el código se ejecuta. – Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla programas que hacen uso de: <ul style="list-style-type: none"> ○ Arreglos ○ Estructuras – Identifica la ejecución del código. – Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
Clases y objetos	35	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado cuando define: <ul style="list-style-type: none"> – Métodos de las clases. – Control de acceso a los miembros de la clase. – Elabora detalladamente un programa orientado a objetos mediante uso de clases. – Emplea jerárquicamente constructores y destructores para la inicialización y destrucción de los objetos, – Verifica minuciosamente que el código se ejecuta correctamente y en caso de errores corrige – Presenta ordenadamente el código fuente y la sintaxis correcta de la programación 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado cuando define: <ul style="list-style-type: none"> – Métodos de las clases. – Control de acceso a los miembros de la clase. – Elabora un programa orientado a objetos mediante uso de clases. – Emplea constructores y destructores para la inicialización y destrucción de los objetos, – Verifica que el código se ejecuta correctamente, y en caso de errores corrige. – Presenta el código fuente y la sintaxis correcta de la aplicación desarrollada. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado cuando define: <ul style="list-style-type: none"> – Métodos de las clases. – Control de acceso a los miembros de la clase. – Elabora un programa orientado a objetos mediante uso de clases. – Emplea constructores y destructores para la inicialización y destrucción de los objetos, – Verifica que el código se ejecuta. – Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado cuando define: <ul style="list-style-type: none"> – Métodos de las clases. – Control de acceso a los miembros de la clase. – Elabora un programa orientado a objetos mediante uso de clases. – Emplea constructores y destructores para la inicialización y destrucción de los objetos, – Identifica la ejecución del código. – Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
		realizada para implementar clases. • Inserta comentarios detallados en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.			
Presentación de resultados	10	• Presenta detalladamente la información más importante descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso y digital. • Elabora minuciosamente el documento con las siguientes características: – Caratula – Objetivo – Contenido – Uso correcto de reglas ortográficas. – Referencias bibliográficas y/o electrónicas empleadas.	• Presenta la información clave descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso y digital. • Elabora el documento con las siguientes características: – Caratula – Objetivo – Contenido – Uso correcto de reglas ortográficas. – Referencias bibliográficas y/o electrónicas empleadas.	• Presenta la información básica descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso o digital. • Elabora el documento omitiendo alguna de las siguientes características: – Caratula – Objetivo – Contenido – Uso correcto de reglas ortográficas. – Referencias bibliográficas y/o electrónicas empleadas.	• Presenta la información mínima requerida descrita en los apartados anteriores en un formato diferente a un organizador gráfico, elaborado en un procesador de texto, en formato impreso o digital. • Elabora el documento con errores, omitiendo dos o más de las siguientes características: – Caratula – Objetivo – Contenido – Uso correcto de reglas ortográficas. – Referencias bibliográficas y/o electrónicas empleadas.

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
Desempeño (Coevaluación)	10	<ul style="list-style-type: none"> Asiste puntualmente a clases y cuenta con el 100% de asistencia. Participa activamente en las actividades de la clase. Muestra perseverancia en el trabajo. Identifica sus errores y planea creativamente nuevas estrategias para mejorar su trabajo. Organiza responsablemente sus tiempos para entregar las actividades en tiempo y forma. Tiene disposición y asume responsablemente el rol asignado en el trabajo colaborativo. 	<ul style="list-style-type: none"> Asiste puntualmente a clases y cuenta con un 90% de asistencias. Participa en las actividades de la clase. Muestra interés en el trabajo. Identifica sus errores y realiza actividades para mejorar su trabajo. Organiza sus tiempos para entregar las actividades. Tiene disposición y asume el rol asignado en el trabajo colaborativo. 	<ul style="list-style-type: none"> Cuenta con un 80% de asistencias. Participa en algunas actividades de la clase. Muestra poco interés en el trabajo. Identifica algunos errores y realiza algunas para mejorar su trabajo. Organiza sus tiempos para entregar algunas actividades. Asume el rol asignado en el trabajo colaborativo con poca disposición. 	<ul style="list-style-type: none"> Cuenta con menos del 80% de asistencias. Evita participar en las actividades de la clase. Muestra resultados ineficientes en el trabajo. Requiere apoyo para identificar sus errores y crear estrategias para mejorar su trabajo. Entrega las actividades fuera de tiempo. Omite el trabajo en equipo.
	100				

Siglema:	PROO-20	Nombre del módulo:	Programación orientada a objetos	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	2.1 Elabora aplicaciones mediante la interacción de los objetos y actores del sistema.		Actividad de evaluación:	2.1.1 Desarrolla programas de cómputo utilizando las propiedades de herencia y polimorfismo.	

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
Herencia	30	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla minuciosamente el código de programación para la implementación de herencia de clases. – Verifica detalladamente que el código se ejecuta correctamente y en caso de errores corrige. – Presenta ordenadamente el código fuente en sintaxis correcta. • Inserta comentarios detallados en el código desarrollado para facilitar 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla el código de programación para la implementación de herencia de clases. – Verifica que el código se ejecuta correctamente y en caso de errores corrige. – Presenta el código fuente en sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla erróneamente el código de programación para la implementación de herencia de clases. – Verifica que el código se ejecuta. – Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla con dificultades el código de programación para la implementación de herencia de clases. – Identifica la ejecución del código. – Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
		la comprensión y la documentación de la programación realizada.			
Sobrecarga de operadores	20	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla minuciosamente el código de programación en el que utilice la sobrecarga de operadores. – Verifica detalladamente que el código se ejecuta correctamente y en caso de errores corrige. – Presenta ordenadamente el código fuente en sintaxis correcta. • Inserta comentarios detallados en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla el código de programación en el que utilice la sobrecarga de operadores. – Verifica que el código se ejecuta correctamente y en caso de errores corrige. – Presenta el código fuente en sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla el código de programación con errores en el que utilice la sobrecarga de operadores. – Verifica que el código se ejecuta. – Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Desarrolla con dificultad el código de programación en el que utilice la sobrecarga de operadores. – Identifica la ejecución del código. – Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
Polimorfismo	30	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Implementa minuciosamente el polimorfismo en clases y funciones virtuales, mediante el desarrollo del código de programación. – Verifica detalladamente que el código se ejecuta correctamente y en caso de errores corrige. – Presenta ordenadamente el código fuente en sintaxis correcta. • Inserta comentarios detallados en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Implementa el polimorfismo en clases y funciones virtuales, mediante el desarrollo del código de programación. – Verifica que el código se ejecuta correctamente y en caso de errores corrige. – Presenta el código fuente en sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Implementa con errores el polimorfismo en clases y funciones virtuales, mediante el desarrollo del código de programación. – Verifica que el código se ejecuta. – Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Implementa con dificultad el polimorfismo en clases y funciones virtuales, mediante el desarrollo del código de programación. – Identifica la ejecución del código. – Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.
Presentación de resultados	10	<ul style="list-style-type: none"> • Presenta detalladamente la información más importante descrita en los apartados anteriores en un organizador gráfico, 	<ul style="list-style-type: none"> • Presenta la información clave descrita en los apartados anteriores en un organizador gráfico, elaborado en un 	<ul style="list-style-type: none"> • Presenta la información básica descrita en los apartados anteriores en un organizador gráfico, elaborado en un 	<ul style="list-style-type: none"> • Presenta la información mínima requerida descrita en los apartados anteriores en un formato diferente a un

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
		<p>elaborado en un procesador de texto, en formato impreso y digital.</p> <ul style="list-style-type: none"> • Elabora minuciosamente el documento con las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<p>procesador de texto, en formato impreso y digital.</p> <ul style="list-style-type: none"> • Elabora el documento con las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<p>procesador de texto, en formato impreso o digital.</p> <ul style="list-style-type: none"> • Elabora el documento omitiendo alguna de las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<p>organizador gráfico, elaborado en un procesador de texto, en formato impreso o digital.</p> <ul style="list-style-type: none"> • Elabora el documento con errores, omitiendo dos o más de las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas.
Disposición en la actividad	10	<ul style="list-style-type: none"> • Asiste puntualmente a clases y cuenta con el 100% de asistencia. • Participa activamente en las actividades de la clase. • Muestra perseverancia en el trabajo. • Identifica sus errores y planea creativamente nuevas estrategias para mejorar su trabajo. • Organiza responsablemente sus tiempos para entregar las actividades en tiempo y forma. 	<ul style="list-style-type: none"> • Asiste puntualmente a clases y cuenta con un 90% de asistencias. • Participa en las actividades de la clase. • Muestra interés en el trabajo. • Identifica sus errores y realiza actividades para mejorar su trabajo. • Organiza sus tiempos para entregar las actividades. • Tiene disposición y asume el rol asignado en el trabajo colaborativo. 	<ul style="list-style-type: none"> • Cuenta con un 80% de asistencias. • Participa en algunas actividades de la clase. • Muestra poco interés en el trabajo. • Identifica algunos errores y realiza algunas para mejorar su trabajo. • Organiza sus tiempos para entregar algunas actividades. • Asume el rol asignado en el trabajo colaborativo con poca disposición. 	<ul style="list-style-type: none"> • Cuenta con menos del 80% de asistencias. • Evita participar en las actividades de la clase. • Muestra resultados ineficientes en el trabajo. • Requiere apoyo para identificar sus errores y crear estrategias para mejorar su trabajo. • Entrega las actividades fuera de tiempo. • Omite el trabajo en equipo.

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
		<ul style="list-style-type: none"> Tiene disposición y asume responsablemente el rol asignado en el trabajo colaborativo. 			
	100				

Siglema:	PROO-20	Nombre del módulo:	Programación orientada a objetos	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	2.2 Asegura la integridad de los datos implementando medidas de seguridad haciendo uso de patrones de diseño.			Actividad de evaluación:	2.2.1 Implementa medidas de seguridad en los programas diseñados haciendo uso de patrones de diseño (Heteroevaluación).

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
Seguridad de la información	30	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Describe detalladamente los siguientes conceptos retomando la información más importante: <ul style="list-style-type: none"> ○ Singular o Singleton. ○ Método de fábrica o Factory Method. ○ Prototipo. ○ Interning. – Implementa minuciosamente la seguridad en las aplicaciones con código de 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Describe los siguientes conceptos retomando la información clave: <ul style="list-style-type: none"> ○ Singular o Singleton. ○ Método de fábrica o Factory Method. ○ Prototipo. ○ Interning. – Implementa la seguridad en las aplicaciones con código de programación, empleando un patrón ordenado en el diseño. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Describe los siguientes conceptos retomando la información básica: <ul style="list-style-type: none"> ○ Singular o Singleton. ○ Método de fábrica o Factory Method. ○ Prototipo. ○ Interning. – Implementa con errores la seguridad en las aplicaciones con código de programación, 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> – Describe los siguientes conceptos retomando la información mínima requerida: <ul style="list-style-type: none"> ○ Singular o Singleton. ○ Método de fábrica o Factory Method. ○ Prototipo. ○ Interning. – Implementa con dificultad la seguridad en las aplicaciones con código de programación,

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
		programación, empleando un patrón jerárquico en el diseño. <ul style="list-style-type: none"> - Verifica detalladamente que el código se ejecuta correctamente y en caso de errores corrige. - Presenta ordenadamente el código fuente en sintaxis correcta. <ul style="list-style-type: none"> • Inserta comentarios detallados en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	<ul style="list-style-type: none"> - Verifica que el código se ejecuta correctamente y en caso de errores corrige. - Presenta el código fuente en sintaxis correcta. <ul style="list-style-type: none"> • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	empleando un patrón en el diseño. <ul style="list-style-type: none"> - Verifica que el código se ejecuta. - Presenta el código fuente en sintaxis. <ul style="list-style-type: none"> • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	omitiendo un patrón en el diseño. <ul style="list-style-type: none"> - Identifica la ejecución del código. - Presenta el código fuente en sintaxis. <ul style="list-style-type: none"> • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.
Cifrado	20	<ul style="list-style-type: none"> • Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> - Realiza detalladamente una investigación, retomando la información más importante sobre algún algoritmo de cifrado de datos y lo 	<ul style="list-style-type: none"> • Emplea el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> - Realiza una investigación retomando los datos clave sobre algún algoritmo de cifrado de datos y lo implementa, haciendo uso del lenguaje. 	<ul style="list-style-type: none"> • Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> - Realiza una investigación retomando los datos básicos sobre algún algoritmo de cifrado de datos y lo implementa con 	<ul style="list-style-type: none"> • Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: <ul style="list-style-type: none"> - Realiza una investigación retomando los datos mínimos requeridos sobre algún algoritmo de cifrado de datos y lo implementa con

INDICADORES	%	C R I T E R I O S			
		Excelente	Bueno	Suficiente	Insuficiente
		implementa jerárquicamente, haciendo uso del lenguaje. - Verifica minuciosamente que el código se ejecuta correctamente y en caso de errores corrige. - Presenta ordenadamente el código fuente en sintaxis correcta. • Inserta comentarios detallados en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada.	- Verifica que el código se ejecuta correctamente y en caso de errores corrige. - Presenta el código fuente en sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada.	algunas fallas, haciendo uso del lenguaje. - Verifica que el código se ejecuta. - Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada.	algunas fallas, omitiendo el uso del lenguaje. - Identifica la ejecución del código. - Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.
Aseguramiento de la operación	30	• Emplea sistemáticamente el lenguaje de programación especificado al realizar las siguientes actividades: - Desarrolla jerárquicamente un código de programación para asegurar la continuidad de la operación en las	• Emplea el lenguaje de programación especificado al realizar las siguientes actividades: - Desarrolla un código de programación para asegurar la continuidad de la operación en las aplicaciones	• Emplea con errores el lenguaje de programación especificado al realizar las siguientes actividades: - Desarrolla con algunas fallas un código de programación para asegurar la continuidad de la operación en las	• Requiere apoyo para emplear el lenguaje de programación especificado al realizar las siguientes actividades: - Desarrolla con dificultad un código de programación para asegurar la continuidad de la operación en las aplicaciones

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
		aplicaciones mediante el manejo detallado de excepciones. <ul style="list-style-type: none"> - Verifica minuciosamente que el código se ejecuta correctamente y en caso de errores corrige. - Presenta ordenadamente el código fuente en sintaxis correcta. • Inserta comentarios detallados en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	mediante el manejo de excepciones. <ul style="list-style-type: none"> - Verifica que el código se ejecuta correctamente y en caso de errores corrige. - Presenta el código fuente en sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y la documentación de la programación realizada. 	aplicaciones mediante el manejo de algunas excepciones. <ul style="list-style-type: none"> - Verifica que el código se ejecuta. - Presenta el código fuente en sintaxis. • Inserta algunos comentarios en el código desarrollado y la documentación de la programación realizada. 	omitiendo el manejo de excepciones. <ul style="list-style-type: none"> - Identifica la ejecución del código. - Presenta el código fuente en sintaxis. • Inserta comentarios básicos en el código desarrollado, omitiendo la documentación de la programación realizada.
Presentación de resultados	10	<ul style="list-style-type: none"> • Presenta detalladamente la información más importante descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso y digital. • Elabora minuciosamente el documento con las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido 	<ul style="list-style-type: none"> • Presenta la información clave descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso y digital. • Elabora el documento con las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido 	<ul style="list-style-type: none"> • Presenta la información básica descrita en los apartados anteriores en un organizador gráfico, elaborado en un procesador de texto, en formato impreso o digital. • Elabora el documento omitiendo alguna de las siguientes características: <ul style="list-style-type: none"> - Caratula - Objetivo - Contenido 	<ul style="list-style-type: none"> • Presenta la información mínima requerida descrita en los apartados anteriores en un formato diferente a un organizador gráfico, elaborado en un procesador de texto, en formato impreso o digital. • Elabora el documento con errores, omitiendo dos o más de las siguientes características:

INDICADORES	%	CRITERIOS			
		Excelente	Bueno	Suficiente	Insuficiente
		<ul style="list-style-type: none"> - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<ul style="list-style-type: none"> - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<ul style="list-style-type: none"> - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas. 	<ul style="list-style-type: none"> - Caratula - Objetivo - Contenido - Uso correcto de reglas ortográficas. - Referencias bibliográficas y/o electrónicas empleadas.
Desempeño de la actividad	10	<ul style="list-style-type: none"> • Asiste puntualmente a clases y cuenta con el 100% de asistencia. • Participa activamente en las actividades de la clase. • Muestra perseverancia en el trabajo. • Identifica sus errores y planea creativamente nuevas estrategias para mejorar su trabajo. • Organiza responsablemente sus tiempos para entregar las actividades en tiempo y forma. • Tiene disposición y asume responsablemente el rol asignado en el trabajo colaborativo. 	<ul style="list-style-type: none"> • Asiste puntualmente a clases y cuenta con un 90% de asistencias. • Participa en las actividades de la clase. • Muestra interés en el trabajo. • Identifica sus errores y realiza actividades para mejorar su trabajo. • Organiza sus tiempos para entregar las actividades. • Tiene disposición y asume el rol asignado en el trabajo colaborativo. 	<ul style="list-style-type: none"> • Cuenta con un 80% de asistencias. • Participa en algunas actividades de la clase. • Muestra poco interés en el trabajo. • Identifica algunos errores y realiza algunas para mejorar su trabajo. • Organiza sus tiempos para entregar algunas actividades. • Asume el rol asignado en el trabajo colaborativo con poca disposición. 	<ul style="list-style-type: none"> • Cuenta con menos del 80% de asistencias. • Evita participar en las actividades de la clase. • Muestra resultados ineficientes en el trabajo. • Requiere apoyo para identificar sus errores y crear estrategias para mejorar su trabajo. • Entrega las actividades fuera de tiempo. • Omite el trabajo en equipo.
	100				