



GOBIERNO DE
MÉXICO

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



Guía pedagógica y de evaluación del módulo

Programación orientada a objetos

Núcleo de Formación Profesional

Área(s):

Tecnología y transporte

Carrera(s):

Profesional Técnico-Bachiller en
Informática

4º semestre

Editor: Colegio Nacional de Educación Profesional Técnica

Módulo: Programación orientada a objetos.

Área(s): Tecnología y transporte.

Carrera(s): PT-B en Informática.

Semestre(s): 4°

Horas por semestre: 144

Créditos por semestre: 14

Fecha de diseño o actualización: 20 de octubre de 2023.

Vigencia: a partir de la aprobación de la junta directiva y en tanto no se genere un documento que lo anule o 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

Manuel de Jesús Espino

Dirección General

Lauro Cordero Frayre

Secretaría General

Hugo Nicolás Pérez González

Secretaría Académica

Edith Chávez Ramos

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 por unidad	9
5 Prácticas y actividades	13
II: Guía de evaluación	
6 Descripción	14
7 Tabla de ponderación	17
8 Matriz de valoración o rúbrica	18

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 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.

2. Generalidades pedagógicas

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 adapten 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.

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.

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-bachiller. 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 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 a comunicar sus ideas. ❖ Aprende a buscar información y a procesarla. ❖ Construye su conocimiento. ❖ Adopta una posición crítica y autónoma. 	<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.

❖ Realiza los procesos de autoevaluación y coevaluación.	❖ 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.
--	---

El docente, en lugar de transmitir vertical y unidireccionalmente los conocimientos, es un mediador del aprendizaje, ya que:

- o 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.
- o Proporciona oportunidades de aprendizaje a los estudiantes apoyándose en metodologías y estrategias didácticas pertinentes a los Resultados de Aprendizaje.
- o Ayuda también al alumno a asumir un rol más comprometido con su propio proceso, invitándole a tomar decisiones.
- o Facilita el aprender a pensar, fomentando un nivel más profundo de conocimiento.
- o Ayuda en la creación y desarrollo de grupos colaborativos entre los alumnos.
- o Guía permanentemente a los alumnos.
- o Motiva al alumno a poner en práctica sus ideas, animándole en sus exploraciones y proyectos.

3. Orientaciones didácticas

Para el desarrollo de las competencias del módulo se recomienda al docente:

- Realizar el encuadre del módulo, tomar acuerdos sobre la forma de trabajar y evaluar con la finalidad de cumplir con las competencias enunciadas en el módulo.
- Definir claramente las actividades y tareas a realizar.
- Fomentar la asistencia a clases.
- Fomentar un ambiente grupal de confianza y respeto para que los alumnos se sientan en libertad de exponer preguntas y/o dudas sobre los contenidos revisados en el módulo.
- Comunicar, escuchar, observar y atender las necesidades educativas y personales del alumno a fin de realimentar su formación académica y reforzar su relación para el desarrollo personal.
- Seleccionar recursos didácticos relacionados con los contenidos enunciados en el módulo.
- Fomentar la democracia y la equidad al tomar acuerdos con el grupo, organizando y dirigiendo situaciones de aprendizaje que promuevan el interés y la participación a través de lo siguiente:
 - Distribución de tareas
 - Preparación de clases con secuencia lógica
 - Proporcionar y recabar información; confiable, relevante y completa
 - Establecimiento de tiempos y formas para el desarrollo de temas y trabajos
- Promover la investigación previa y permanente, lo que permitirá al alumno participar activamente durante el desarrollo de temas y potenciará su habilidad para realizar análisis crítico de los materiales bibliográficos para construir nuevos conocimientos y aprendizajes significativos.
- Fomentar el trabajo individual, por equipo y grupal con la finalidad de promover la generación de nuevas ideas y el trabajo colaborativo.
- Utilizar experiencias personales y profesionales en el campo laboral, ejercicios, ejemplos de casos reales, etc., que le permitan al alumno relacionar aprendizajes previos con nuevos.
- Plantear casos prácticos en los que el alumno pueda poner en práctica lo aprendido en el módulo.
- Administrar la progresión de los aprendizajes, mediante el registro de avances y dificultades durante el desarrollo del programa, como un medio de guiar la realimentación.
- Considerar los tres tipos de evaluación: diagnóstica, formativa y sumativa.

4. Estrategias de aprendizaje por unidad

Unidad: 1. Manejo de componentes del lenguaje orientados a objetos.

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

- Realizar una investigación documental a través de la consulta de las fuentes sugeridas o haciendo uso de internet, para analizar y describir la metodología de la programación orientada a objetos y los elementos empleados en desarrollo de aplicaciones, describiendo los siguientes:
 - 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
- Realiza el modelado de diagrama de casos de uso de una tienda de electrodomésticos.
- Desarrolla el modelado de diagrama de casos de uso de una cadena de videoclubes.
- Elabora el modelado de diagrama de casos de uso de una máquina expendedora y de venta de bebidas.
- Realiza el modelado de diagrama de clases del evento de la inscripción escolar.
- Desarrolla el modelado de diagrama de clases de un estacionamiento público.
- Elabora el modelado de diagrama de secuencia de un centro de instalaciones deportivas.
- Realiza el modelado de diagrama de secuencia de un inicio de sesión en un sistema.
- Realizar una investigación documental, para analizar y describir 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 alumno aplicar las siguientes estructuras manejadas en el lenguaje de programación seleccionado por el maestro:

- Aplicar las funciones de entrada y salida estándar.
- Aplicar las siguientes estructuras de decisión, para definir el orden de ejecución de bloques de código:
 - if – else
 - else – if
 - switch
- Aplicar las siguientes estructuras de ciclo para la ejecución continua y/o repetida de bloques de código.
 - While
 - Do – while
 - For
 - Ciclos infinitos.
- Utilizar los siguientes tipos de datos estructurados:
 - Arreglos
 - Estructuras
- Implementar clases basado en POO
- Desarrollar 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
- Aplicar las siguientes sentencias en el manejo de errores de excepción.
 - class
 - private
 - public
 - protected
- **Realizar la actividad de evaluación 1.2.1 considerando la rúbrica correspondiente**

Unidad: 2. Desarrollo de aplicaciones orientadas a objetos

Para el desarrollo del resultado de aprendizaje **2.1**, se recomienda al alumno aplicar los siguientes elementos de la programación orientada a objetos en el lenguaje de programación seleccionado por el maestro:

- Implementar el manejo de herencia basado en POO.
- Implementar la sobrecarga de operadores.
- Implementar la sobrecarga de operadores + y - basado en POO
- Implementar la sobrecarga de operadores +, ==, ++ y << basado en POO.
- Implementar Suma y resta de números complejos basado en POO.
- Implementar el 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 alumno implementar los siguientes conceptos y elementos de seguridad aplicados en la implementación de programas orientados a objetos utilizando el lenguaje de programación seleccionado por el maestro:

- Realizar una investigación documental, para analizar y describir los patrones de diseño más comunes empleados para la autenticación de aplicaciones orientadas a objetos, en el que describa:
 - Fabricas abstractas.
 - Singleton o Singular.
 - Factory Method.
 - Prototipo.

- Realizar una investigación documental acerca de algún algoritmo de cifrado de datos y lo implementa en alguna aplicación.
- Aplicar las siguientes sentencias en el manejo de errores de excepción.
 - try
 - catch
 - throw
- Implementar en un programa el manejo de errores de excepción basado en POO
- Implementar en un programa el manejo de excepciones en clases basado en POO.
- Implementar 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 libertad de cátedra, este apartado quedará bajo la responsabilidad de los docentes para que, de acuerdo con su experiencia, las características del grupo y el desempeño de los estudiantes, seleccione, proponga y realice aquellas que garanticen un mayor desarrollo de competencias, 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 docente implica, ante todo, una labor de investigación y promoción del autoaprendizaje para ofrecer a los educandos la información más actualizada, así como las actividades que permitan un mayor logro de los objetivos educacionales, considerando las características del grupo y del contexto en donde se desarrolla el proceso de enseñanza-aprendizaje, ya sea en el sistema presencial o en el mixto.

En este sentido, se confía en el docente como un líder que fomenta la creatividad y el emprendimiento, considerando que el aprendizaje se dará de mejor manera si el alumno relaciona la teoría con la vida diaria, con la resolución de problemas, brindando las bases científicas de la práctica, a fin de transformar el mundo concreto.

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

Derivado de lo anterior, para promover en los alumnos el “saber hacer” integrando conocimientos, habilidades y actitudes, se sugiere la planeación de actividades y prácticas que vayan de lo más simple a lo más complejo, de lo conocido a lo desconocido, en escenarios lo más reales posible, para alcanzar los logros establecidos en los Resultados de Aprendizaje y con ello, lograr la vinculación de la teoría con la práctica.

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 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.

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 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.

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 las mismas personas.

La **coevaluación** en la que los alumnos se evalúan mutuamente, es decir, evaluadores y evaluados intercambian su papel alternativamente; 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 alumno y al docente:

- 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 se orienten hacia 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** que 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.

De acuerdo con lo anterior, en las rúbricas se sugiere el momento para que se lleven a cabo estas 3 modalidades de evaluación: un indicador para que los alumnos practiquen la auto y la coevaluación, y una actividad de evaluación para que un docente externo al grupo evalúe el desempeño del alumno a través de la rúbrica.

Cada uno de los Resultados de Aprendizaje (RA) tiene asignada al menos una actividad de evaluación (AE), a la cual 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, 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 dicha actividad con respecto al RA y éste a su vez, con respecto a la Unidad de Aprendizaje.

La ponderación que se asigna en cada una de las actividades queda 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. 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 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. Una 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.

Asimismo, es necesario que el docente realice la captura de la evaluación de los resultados de aprendizaje en el Sistema de Administración Escolar (SAE), considerando las fechas de corte establecidas en el calendario escolar del Sistema CONALEP, a fin de no afectar el desempeño de los alumnos y disminuir los índices de reprobación y abandono escolar.

7. Tabla de ponderación

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

Siglema:	PROO -04	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	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Elementos de la programación orientada a objetos	30	<ul style="list-style-type: none"> •Elabora una investigación documental acerca de la programación orientada a objetos y sus elementos, en la que describe: <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 mapa conceptual para describir la forma en que interactúan los diferentes elementos de la Programación orientada a objetos. 	Elabora una investigación documental acerca de la programación orientada a objetos y sus elementos, en la que describe: <ul style="list-style-type: none"> - Programación orientada a objetos. - Objetos. - Clases. - Modelo de clases. - Abstracción. - Encapsulación. - Herencia. - Polimorfismo. -Relación entre clases y objetos. 	Omite describir alguno de los siguientes elementos. <ul style="list-style-type: none"> - Programación orientada a objetos. - Objetos. - Clases. - Modelo de clases. - Abstracción. - Encapsulación. - Herencia. - Polimorfismo. - Relación entre clases y objetos.

<p>Lenguaje de modelado</p>	<p>30</p>	<ul style="list-style-type: none"> • Elabora modelos orientados a objetos, haciendo uso de los siguientes tipos de diagramas: <ul style="list-style-type: none"> - Cajas de uso - Clases - Actividad. • Presenta en orden 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 modelos orientados a objetos, haciendo uso de los siguientes tipos de diagramas: <ul style="list-style-type: none"> - Cajas de uso - Clases - actividad. • Presenta en orden los diagramas elaborados. 	<p>Omite alguna de las siguientes características:</p> <ul style="list-style-type: none"> • Elabora modelos orientados a objetos, haciendo uso de los siguientes tipos de diagramas: <ul style="list-style-type: none"> - Cajas de uso - Clases - Actividad.
<p>Software de desarrollo</p>	<p>30</p>	<ul style="list-style-type: none"> • Elabora una tabla para describir al menos 3 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos en el que describa: <ul style="list-style-type: none"> – Plataforma. – Editor de texto empleado. – Compilador. – Enlazador. – Entornos de desarrollo integrados. • Presenta en orden la tabla elaborada. • Presenta un análisis de ventajas de portabilidad entre las diferentes plataformas que presentan algunos lenguajes de programación orientados a objetos. 	<ul style="list-style-type: none"> • Elabora una tabla para describir al menos 3 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos en el que describa: <ul style="list-style-type: none"> – Plataforma. – Editor de texto empleado. – Compilador. – Enlazador. – Entornos de desarrollo integrados. • Presenta en orden la tabla elaborada. 	<p>Excluye alguna condición:</p> <ul style="list-style-type: none"> • Elabora una tabla para describir al menos 3 lenguajes de programación que proveen soporte para el desarrollo de aplicaciones orientadas a objetos en el que describa: <ul style="list-style-type: none"> – Plataforma. – Editor de texto empleado. – Compilador. – Enlazador. – Entornos de desarrollo integrados. • Presenta en orden la tabla elaborada,
		<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), 	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital. 	<p>Incumple con alguna actividad:</p> <ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un

<p>Presentación de resultados (Autoevaluación)</p>	<p>10</p>	<p>con estructura (caratula, objetivo y contenido,)</p> <ul style="list-style-type: none"> • Cumple con los criterios de contenido y presentación establecidos por el docente • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas. • Presenta el documento impreso limpio, respetando la estructura definida para su diseño. 	<ul style="list-style-type: none"> • Cumple con los criterios de contenido y presentación establecidos por el docente • Redacta documento aplicando las reglas ortográficas y gramaticales • Incluye referencias documentales y/o electrónicas empleadas. • Redacta documento aplicando las reglas ortográficas y gramaticales. 	<p>procesador de texto y/o presentador gráfico),</p> <ul style="list-style-type: none"> • Estructura (caratula, objetivo y contenido) • Redacta documento aplicando las reglas ortográficas y gramaticales. • Referencias documentales y/o electrónicas empleadas.
<p>100</p>				

Siglema:	PROO-04	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.

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Funciones de entrada y salida	15	Emplea lenguaje de programación especificado por el docente, cuando: <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 orden y sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	Emplea lenguaje de programación especificado por el docente, cuando: <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 orden y sintaxis correcta. 	Omite alguna característica al emplear el lenguaje de programación especificado por el docente, cuando: <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 orden y sintaxis correcta. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.
Sentencias de control	15	Emplea lenguaje de programación especificado por el docente, cuando: <ul style="list-style-type: none"> • Desarrolla programas utilizando sentencias de control: 	Emplea lenguaje de programación especificado por el docente, cuando: <ul style="list-style-type: none"> • Desarrolla programas utilizando sentencias de control: <ul style="list-style-type: none"> ○ If else ○ else if 	Omite alguna característica al emplear lenguaje de programación especificado por el docente, cuando: <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 <ul style="list-style-type: none"> • Verifica que el código se ejecuta correctamente, y en caso de errores corrige. • Presenta el código fuente en orden y sintaxis correcta de la aplicación desarrollada. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	<ul style="list-style-type: none"> • switch • for • while • do while • Verifica que el código se ejecuta correctamente, y en caso de errores corrige • Presenta el código fuente en orden y sintaxis correcta de la aplicación desarrollada. 	<ul style="list-style-type: none"> ○ If else ○ else if ○ switch ○ for ○ while ○ do while <ul style="list-style-type: none"> • Verifica que el código se ejecuta correctamente, y en caso de errores corrige. • Presenta el código fuente en orden y sintaxis correcta de la aplicación desarrollada. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.
Arreglos y estructuras	15	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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 en orden y sintaxis correcta de la aplicación desarrollada. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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 en orden y sintaxis correcta de la aplicación desarrollada. 	<p>Omite alguna característica al emplear lenguaje de programación especificado por el docente, cuando:</p> <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 en orden y sintaxis correcta de la aplicación desarrollada. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>

<p>Clases y objetos</p>	<p>35</p>	<p>Emplea lenguaje de programación especificado por el docente, cuando define:</p> <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 en orden y sintaxis correcta de la programación realizada para implementar clases. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	<p>Emplea lenguaje de programación especificado por el docente, cuando define:</p> <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, • Presenta el código fuente en orden y sintaxis correcta de la programación realizada para implementar clases. 	<p>Omite alguna característica al emplear lenguaje de programación especificado por el docente, cuando define:</p> <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 en orden y sintaxis correcta de la programación realizada para implementar clases. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
<p>Presentación de resultados</p>	<p>10</p>	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,) 	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,) • Cumple con los criterios de contenido y presentación establecidos. 	<p>Omite 2 o más de las siguientes características:</p> <ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,)

		<ul style="list-style-type: none"> • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas. • Presenta el documento impreso limpio, respetando la estructura definida para su diseño. 	<ul style="list-style-type: none"> • Redacta documento aplicando las reglas ortográficas y gramaticales. 	<ul style="list-style-type: none"> • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas. • Presenta el documento impreso limpio, respetando la estructura definida para su diseño.
Desempeño (Coevaluación)	10	<ul style="list-style-type: none"> • Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Tiene disposición y asume rol asignado en el trabajo colaborativo 	<ul style="list-style-type: none"> • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente • Muestra disposición y asume rol asignado en el trabajo colaborativo 	<p>Omite 2 o más de las siguientes características:</p> <ul style="list-style-type: none"> • Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Tiene disposición y asume rol asignado en el trabajo colaborativo
	100			

Siglema:	PROO-04	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	Suficiente	Insuficiente
Herencia	30	Emplea lenguaje de programación especificado por el docente, cuando: <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 orden y sintaxis correcta de la implementación de herencia de clases. Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.	Emplea lenguaje de programación especificado por el docente, cuando: <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 orden y sintaxis correcta de la implementación de herencia de clases. 	Omite dos o más características al emplear lenguaje de programación especificado por el docente, cuando: <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 orden y sintaxis correcta de la implementación de herencia de clases.
Sobrecarga de operadores	20	Emplea lenguaje de programación especificado por el docente, cuando:	Emplea lenguaje de programación especificado por el docente, cuando: <ul style="list-style-type: none"> Desarrolla el código de programación en el que utilice la sobrecarga de operadores. 	Omite dos o más características al emplear lenguaje de programación especificado por el docente, cuando:

		<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 orden y sintaxis correcta del uso de sobrecarga de operadores. • Inserta comentarios en el código desarrollado para facilitar la comprensión y 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 orden y sintaxis correcta del uso de sobrecarga de operadores. 	<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 orden y sintaxis correcta del uso de sobrecarga de operadores.
Polimorfismo	30	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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 orden y sintaxis correcta de la implementación de polimorfismo. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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 orden y sintaxis correcta de la implementación de polimorfismo. 	<p>Omite dos o más características al emplear lenguaje de programación especificado por el docente, cuando:</p> <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 orden y sintaxis correcta de la implementación de polimorfismo.

<p>Presentación de resultados</p>	<p>10</p>	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,) • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas. <p>Presenta el documento impreso limpio, respetando la estructura definida para su diseño.</p>	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital. • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. <p>Incluye referencias documentales y/o electrónicas empleadas</p>	<p>Incumple con alguna o varias de las siguientes actividades:</p> <ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,) • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas.
<p>Disposición en la actividad</p>	<p>10</p>	<ul style="list-style-type: none"> • Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Tiene disposición y asume rol asignado en el trabajo colaborativo 	<ul style="list-style-type: none"> • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente • Muestra disposición y asume rol asignado en el trabajo colaborativo 	<p>Carece de lo siguiente:</p> <ul style="list-style-type: none"> • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Muestra disposición y asume rol asignado en el trabajo colaborativo
<p>100</p>				

Siglema:	PROO-04	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	Suficiente	Insuficiente
Seguridad de la información	30	Emplea lenguaje de programación especificado por el docente, cuando: <ul style="list-style-type: none"> Describe: <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 de diseño. Verifica que el código se ejecuta correctamente y en caso de errores corrige. Presenta el código fuente en orden y sintaxis correcta del patrón de diseño implementado. Inserta comentarios en el código desarrollado para 	Emplea lenguaje de programación especificado por el docente, cuando <ul style="list-style-type: none"> Describe: <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 de diseño. Verifica que el código se ejecuta correctamente y en caso de errores corrige. 	Carece de alguna o varias de las siguientes características: <ul style="list-style-type: none"> Describe: <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 de diseño. Verifica que el código se ejecuta correctamente y en caso de errores corrige. Presenta el código fuente en orden y sintaxis correcta del patrón de diseño implementado.

		facilitar la comprensión y documentación de 3ra programación realizada.		
Cifrado	20	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> Realiza una investigación sobre algún algoritmo de cifrado de datos y lo implementa, haciendo uso del lenguaje. Verifica que el código se ejecuta correctamente y en caso de errores corrige. Presenta el código fuente en orden y sintaxis correcta del cifrado de datos. Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> Realiza una investigación sobre algún algoritmo de cifrado de datos y lo implementa, haciendo uso del lenguaje. Presenta el código fuente en orden y sintaxis correcta del cifrado de datos. Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	<p>Omite alguna o varias de las siguientes características:</p> <ul style="list-style-type: none"> Realiza una investigación sobre algún algoritmo de cifrado de datos y lo implementa, haciendo uso del lenguaje. Verifica que el código se ejecuta correctamente y en caso de errores corrige. Presenta el código fuente en orden y sintaxis correcta del cifrado de datos. Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.
Aseguramiento de la operación	30	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> Desarrolla código de programación para asegurar la continuidad de la operación en las aplicaciones mediante el manejo de excepciones. Verifica que el código se ejecuta correctamente y en caso de errores corrige. 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> Desarrolla código de programación para asegurar la continuidad de la operación en las aplicaciones mediante el manejo de excepciones. Presenta el código fuente en orden y sintaxis correcta del manejo de excepciones. Inserta comentarios en el código desarrollado para facilitar la comprensión y 	<p>Omite alguna o varias de las siguientes características:</p> <ul style="list-style-type: none"> Desarrolla código de programación para asegurar la continuidad de la operación en las aplicaciones mediante el manejo de excepciones. Verifica que el código se ejecuta correctamente y en caso de errores corrige. Presenta el código fuente en orden y sintaxis correcta del manejo de excepciones.

		<ul style="list-style-type: none"> • Presenta el código fuente en orden y sintaxis correcta del manejo de excepciones. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 	documentación de la programación realizada.	
Presentación de resultados	10	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,) • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas. Presenta el documento impreso limpio, respetando la estructura definida para su diseño. 	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital. • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas. 	<p>Incumple con alguna o varias de las siguientes actividades:</p> <ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,) • Cumple con los criterios de contenido y presentación establecidos. • Redacta documento aplicando las reglas ortográficas y gramaticales. • Incluye referencias documentales y/o electrónicas empleadas.
Desempeño de la actividad	10	<ul style="list-style-type: none"> • Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades 	<ul style="list-style-type: none"> • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. 	<p>Carece de lo siguiente:</p> <ul style="list-style-type: none"> • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades

		<p>previas para mejorar su trabajo.</p> <ul style="list-style-type: none"> • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Tiene disposición y asume rol asignado en el trabajo colaborativo. 	<ul style="list-style-type: none"> • Muestra responsabilidad al entregar en la fecha establecida por el docente • Muestra disposición y asume rol asignado en el trabajo colaborativo. 	<p>previas para mejorar su trabajo.</p> <ul style="list-style-type: none"> • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Muestra disposición y asume rol asignado en el trabajo colaborativo.
100				