

Guía Pedagógica y de Evaluación del Módulo Programación orientada a objetos

I. Guía Pedagógica del Módulo Programación orientada a objetos

Editor: Colegio Nacional de Educación Profesional Técnica

Guía pedagógica y de evaluación del Módulo: Programación orientada a objetos.

Área(s): Tecnología y transporte.

Carrera(s): Profesional Técnico y Profesional Técnico –Bachiller en Informática.

Semestre(s): Cuarto.

© Colegio Nacional de Educación Profesional Técnica

Fecha de diseño o actualización: 30 de noviembre de 2017..

Vigencia: Dos años, en tanto no se produzca un documento que lo anule o desaparezca el objeto del actual.

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

Directorio

Director General

Jorge Alejandro Neyra Gonzalez

Secretario General

Jorge Galileo Castillo Vaquera

Secretaria Académica

María Elena Salazar Peña

Secretaria de Administración

Aida Margarita Ménez Escobar

Secretario de Planeación y Desarrollo Institucional

Agustín Arturo González de la Rosa

Secretario de Servicios Institucionales

Javier Rodrigo Villegas Garcés

Director Corporativo de Asuntos Jurídicos

Iván Hernán Sierra Santón

Titular de la Unidad de Estudios e Intercambio Académico

Patricia Guadalupe Guadarrama Hernández

Director Corporativo de Tecnologías Aplicadas

Iván Flores Benítez

Directora de Diseño Curricular

Marisela Zamora Anaya

Coordinadora de las Áreas Básicas y de Servicios

Caridad del Carmen Cruz López

Coordinador de las Áreas de Mantenimiento e Instalación,
Electricidad, Electrónica y TIC

Nicolás Guillermo Pinacho Burgoa

Coordinadora de las Áreas de Procesos de Producción y
Transformación

Norma Osorio Vera

Grupo de trabajo:

Actualiza:

Técnico:

Con la asesoría de consultores contratados

Metodológico

Con la asesoría de consultores contratados

Contenido

	Pág.
I. Guía pedagógica	
1. Descripción	6
2. Datos de identificación del estándar de competencia	7
3. Generalidades pedagógicas	8
4. Enfoque del módulo	10
5. Orientaciones didácticas y estrategias de aprendizaje por unidad	11
6. Prácticas/Actividad	15
II. Guía de evaluación	67
7. Descripción	68
8. Tabla de ponderación	71
9. Desarrollo de actividades de evaluación	72
10. Matriz de valoración o rúbrica	74

1. Descripción

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

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

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

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

2. Datos de Identificación del estándar de competencia

Título:			
Código:		Nivel de competencia:	
Elementos de competencia laboral			

3. Generalidades pedagógicas

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, apoyarse mutuamente y establecer relaciones positivas y de confianza.

Por otro lado, 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 correspondientes a la formación básica, propedéutica y profesional.

Los principios asociados a la concepción constructivista del aprendizaje mantienen una estrecha relación con los de la educación basada en competencias, la cual se ha concebido en el Colegio como el enfoque idóneo para orientar la formación ocupacional de los futuros profesionales técnicos y profesional 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 comunica sus ideas. ❖ Aprende a buscar información y a procesarla. ❖ Construye su conocimiento. ❖ Adopta una posición crítica y autónoma. ❖ Realiza los procesos de autoevaluación y coevaluación. 	<ul style="list-style-type: none"> ❖ Organiza su formación continua a lo largo de su trayectoria profesional. ❖ Domina y estructura los saberes para facilitar experiencias de aprendizaje significativo. ❖ Planifica los procesos de enseñanza y de aprendizaje atendiendo al enfoque por competencias, y los ubica en contextos disciplinares, curriculares y sociales amplios. ❖ Lleva a la práctica procesos de enseñanza y de aprendizaje de manera efectiva, creativa e innovadora a su contexto institucional. ❖ Evalúa los procesos de enseñanza y de aprendizaje con un enfoque formativo. ❖ Construye ambientes para el aprendizaje autónomo y colaborativo. ❖ Contribuye a la generación de un ambiente que facilite el desarrollo sano e integral de los estudiantes. ❖ Participa en los proyectos de mejora continua de su escuela y apoya la gestión institucional.

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

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

4. Enfoque del Módulo

El módulo de **Programación Orientada a Objetos** está enfocado en la adquisición de competencias necesarias para desarrollar aplicaciones bajo el enfoque de la programación orientada a objetos empleando un lenguaje de programación de tercera generación. Se construyen modelos para el diseño de programas orientados a objetos empleando el Lenguaje de Modelado Unificado o UML, y se desarrollan aplicaciones, empleando las técnicas de herencia, sobrecarga, polimorfismo y encapsulamiento, así como la implementación de medidas de seguridad para la integridad de los datos y la ejecución continua del sistema.

Sin dejar de poner énfasis en que la enseñanza debe personalizarse, en el sentido de permitir a cada alumno trabajar con independencia y a su propio ritmo, también es importante incluir y promover en este módulo estrategias de aprendizaje colaborativo y grupal, así como también fomentar el desarrollo de competencias transversales que permitan establecer una mejor comunicación e interrelaciones con los demás, socializar, compartir e intercambiar información, potencializar un pensamiento crítico, lo que contribuye a activar el aprendizaje y autoaprendizaje.

El contexto en el que se estructura el módulo parte de la consideración de que en la sociedad actual se exige que la escuela “forme” y “prepare para la vida”, de modo que el alumno no sólo responda a situaciones inmediatas, sino que adquiera conocimientos, habilidades y actitudes que le permitan una participación reflexiva y comprometida con su entorno local y mundial. Para lograrlo hay que enfatizar el desarrollo de competencias que permitan a los sujetos comprender el mundo e insertarse de manera exitosa a la sociedad,

El desarrollo de estas competencias implica interrelaciones mutuas. Así, el promover la cultura del respeto y la solidaridad se vincula fácilmente con la capacidad de tomar decisiones y la iniciativa de llevar a cabo proyectos personales, aprovechando diversas informaciones y los avances de la ciencia. El reto docente es cómo integrar las competencias y favorecer el desarrollo de las disciplinas que pretenda enseñar ya que el enfoque de competencias que se plantea no alude únicamente a las “competencias para la vida”, sino a las competencias básicas, disciplinares y profesionales que forman un todo.

En lo que se refiere al aprendizaje procedimental, implica la consecución del propósito del módulo a través de acciones secuenciadas que lleven gradualmente al alumno al desarrollo de sus actividades, de manera segura, consciente y responsable.

Se recomienda por último elaborar un código ético durante el desarrollo del módulo con el propósito de definir los compromisos y responsabilidades que deben compartir en el espacio académico, como: respeto a la persona, honestidad, confianza, justicia, comunicación, cooperación, iniciativa, amabilidad, perseverancia y la actitud positiva para el logro de objetivos, así como adecuar las prácticas de ejercicio al equipo existente en el laboratorio de informática y al área de aplicación de la carrera; realizando las prácticas con orden, limpieza, fomentando el uso de software libre o de marca, evitando acciones ilegales para garantizar el funcionamiento y calidad del mismo.

5. Orientaciones didácticas y estrategias de aprendizaje por unidad

Unidad I:	Manejo de componentes del lenguaje orientados a objetos.
Orientaciones Didácticas	

Para el desarrollo de la presente unidad se recomienda al docente:

- Aplicar una evaluación diagnóstica al iniciar la unidad, definiendo los contenidos y resultados de aprendizaje a alcanzar, también se exhorta a establecer reglas grupales para el desarrollo de las clases y acuerdos en el cumplimiento de las tareas encomendadas dentro y fuera del aula, y por último es relevante promover la asistencia y participación diaria, el cuidado de los materiales, recursos didácticos y equipo de trabajo.
- Abordar la unidad con ejercicios, exposiciones, demostraciones, simulaciones, prácticas, comentarios, conclusiones, recopilaciones y entrega de evidencias en forma individual, en dúos, en equipos y en grupo para favorecer hábitos de estudio, trabajo personal, en equipo, unión y trabajo colaborativo. Lo anterior con el fin de hacer significativo el aprendizaje del alumno en el manejo de la terminología técnica e impulsar el desarrollo de habilidades en el uso de la programación orientada a objetos.
- Abordar el diseño de programas bajo el enfoque de la metodología orientada a objetos, a través de la elaboración de diagramas de uso, diagramas de clases y de actividad, para mostrar todas las piezas que integran la solución y la manera en cómo se relacionan, a fin de tener una visión más amplia de lo que se requiere obtener.
- Desarrollar programas para el manejo de funciones de entrada y salida, manejo de sentencias de control, métodos de ordenamiento, búsqueda de datos, definición e implementación de clases y el control de acceso a los miembros de una clase, empleando un lenguaje de programación de tercera generación.

Fortalece las siguientes competencias transversales:

- 7.3 Subrayar la importancia que tiene la presencia del alumno en cada clase, su participación para el enriquecimiento del aprendizaje de todo el grupo y la asignación de tareas y actividades intra y extramuros, con el fin de incentivar en él su cumplimiento voluntario y oportuno.
- 5.2 Organizar sistemáticamente la información que se ha de manejar y procesar para su aprendizaje. Efectuando explícitamente la vinculación de esta unidad con la que precede, con el fin de que el alumno valore su importancia académica y curricular.

8.3 Promover la dinámica grupal colaborativa y cooperativa a través de la realización de las técnicas didácticas y de aprendizaje correspondientes, durante el transcurso de cada sesión para favorecer el clima que fomente el intercambio constructivo de ideas.

Estrategias de Aprendizaje	Recursos didácticos
<ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> - Programación orientada a Objetos. - Objetos. - Mensajes. - Paso de mensajes. - Estructura interna de los objetos. - Herencia. - Polimorfismo. - Clases. - Diferencia entre clases y objetos. <p>Lenguaje de Modelado</p> <ul style="list-style-type: none"> • Resolver el problema No. 1. Modelado de diagrama de casos de uso de una tienda de electrodomésticos. • Resolver el problema No. 2. Modelado de diagrama de casos de uso de una cadena de videoclubes. • Resolver el problema No. 3. Modelado de diagrama de casos de uso de una máquina expendedora y de venta de bebidas. • Resolver el problema No. 4 Modelado de diagrama de clases del evento del comité olímpico Internacional. • Resolver el problema No. 5 Modelado de diagrama de clases de un estacionamiento. • Resolver el problema No. 6 Modelado de diagrama de secuencia de un centro de instalaciones deportivas. • Realizar el problema No. 7 Modelado de diagrama de secuencia de un controlador de pistas scalestrix. • 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: <ul style="list-style-type: none"> - Plataforma - Editor de texto expleado. 	<p>Básicas:</p> <ul style="list-style-type: none"> • Ceballos, F. (2015). <i>C/C++ - Curso de programación 4ta Ed.</i> Paracuellos del Jarama, Madrid, España. RA-MA. • Joyanes, L. (1996). <i>Programación Orientada a Objetos.</i> Aravaca, Madrid, España. McGraw-Hill • Schmuller, J. (2001). <i>Aprendiendo UML en 24 horas.</i> Naucalpan de Juárez, México, Prentice Hall. <p>Complementaria:</p> <ul style="list-style-type: none"> • Kimmel, Paul. (2008). <i>Manual UML.</i> México. McGraw-Hill Interamericana <p>Páginas web:</p> <ul style="list-style-type: none"> • SoloLearn. <i>Curso de programación C++.</i> https://play.google.com/store/apps/details?id=com.sololearn.cplusplus • Ruiz, R. <i>Fundamentos de Programación Orientada a Objetos.</i> Recuperado el 21/10/2017 de https://goo.gl/81KtA5 • C++ Con Clase. <i>C++ Con Clase.</i> Recuperado el 21/10/2017 de http://c.conclase.net/

- Compilador.
- Enlazador.
- Entornos de Desarrollo Integrado.

- **Realizar la actividad de evaluación 1.1.1** Donde “*Modelaje bajo el enfoque de la programación orientada a objetos*”
 - Tabla descriptiva de características de lenguajes de programación orientada a objeto.
 - Diagramas de modelos para el diseño orientado a objeto
- Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos

Funciones de entrada y salida.

- Aplicar las siguientes funciones de entrada y salida estándar:
 - `cout <<`
 - `cin >>`
- **Realiza las prácticas No. 1 y 2** Uso de funciones de entrada y salida estándar basado en P00.
- Aplicar las siguientes estructuras de decisión para definir el orden de ejecución de bloques de código:
 - `if – else`
 - `else – if`
 - `switch`
- **Realizar la práctica No. 3** Uso de sentencia `if-else` basado en P00
- **Realizar la práctica No. 4** Uso de sentencia `else-if` basado en P00
- **Realizar la práctica No. 5** Uso de sentencia `switch` basado en P00
- Aplicar las siguientes estructuras de ciclo para la ejecución continua y/o repetida de bloques de código.
 - `While`
 - `Do – while`
 - `For`
 - `Ciclos infinitos.`
- **Realizar la práctica No. 6** Uso de sentencia bucle `while` basado en P00
- **Realizar la práctica No. 7** Uso de sentencia bucle `do ... while` basado en P00
- **Realizar la práctica No. 8** Uso de sentencia bucle `for` basado en P00

Arreglos y estructuras

- **Realizar la práctica No. 9** Uso de arreglos

- ArgoUML. *ArgoUML*. Recuperado el 21/10/2017 de <http://argouml.tigris.org/>
- Microsoft. *Microsoft Visio*. Recuperado el 21/10/2017 de <https://goo.gl/1CTNIE>
- UML, *UML*. Recuperado el 21/10/2017 de <http://www.uml.org/>

- Realizar la práctica No. 10 Uso de estructuras

Clases y objetos.

- Realizar la práctica No. 11, 12 y 13 Implementar clases basado en P00
- Realizar la 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
 - Arreglos y estructuras
 - Clases y objetos
- Aplicar las siguientes sentencias en el manejo de errores de excepción.
 - *class*
 - *private*
 - *public*
 - *protected*

Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos

Unidad II:	Desarrollo de aplicaciones orientadas a objetos
Orientaciones Didácticas	

Para el desarrollo de la presente unidad se recomienda al docente:

- Definir los contenidos y resultados de aprendizaje a alcanzar, realizar ejercicios, demostraciones, simulaciones, prácticas, comentarios, conclusiones, recapitulaciones y entrega de evidencias mediante dinámicas en forma individual, en dúos, en equipos y en grupo para favorecer hábitos de estudio, trabajo personal, en equipo, trabajo colaborativo e intercambio constructivo de ideas.
- Enfocar los conceptos centrales, en el sustento teórico metodológico y los principios de cada tópico a desarrollar aplicados al conocimiento y la resolución de problemas, así como supervisar, apoyar, resolver dudas y observaciones planteadas por los alumnos durante el desarrollo de la unidad, los ejercicios exploratorios y las prácticas de ejercicios. Hacer hincapié en explicar y ejemplificar las características de Abstracción, Encapsulamiento, Polimorfismo y Herencia, así como el manejo de excepciones para evitar la caída del sistema y asegurar la continuidad de operación.
- Abordar el desarrollo de programas de cómputo para la interacción de los objetos y actores del sistema, mediante la identificación e implementación de las propiedades de la herencia para la reutilización de código, el uso del polimorfismo en funciones y la sobrecarga de operadores para definir clases diferentes que tienen métodos o atributos denominados de forma idéntica, pero que se comportan de manera distinta.
- Aplicar medidas de seguridad para la conservación de la integridad de los objetos y sus datos, así como medidas para la operación continua del sistema mediante la implementación de patrones de diseño de autenticación.

Fortalecer las siguientes competencias transversales:

- 7.3 Organizar sistemáticamente la información que se ha de manejar y procesar para su aprendizaje. Efectuando explícitamente la vinculación de esta unidad con la que precede.
- 8.4 Proponer planteamientos de problemas y su solución, mediante la elaboración de ejercicios de análisis de casos reales aplicados.
- 6.1 Sugerir al alumno realizar consultas de información de sitios que sean confiables y leer las reglas de uso de los sitios web consultados.
- 7.1 Efectuar el proceso de evaluación continua que haga referencia al proceso sistemático y permanente mediante el cual se haya valorado el logro de los objetivos planteados y el desarrollo de resolución de problemas por parte del alumno.

Estrategias de Aprendizaje	Recursos didácticos
<p>Herencia</p> <ul style="list-style-type: none"> Realizar las prácticas No. 14 y 15 Manejo de herencia basado en P00. <p>Sobrecarga de operadores</p> <ul style="list-style-type: none"> Realizar la práctica No. 16 Sobrecarga de operadores + y - basado en P00 Realizar la práctica No. 17 Sobrecarga de operadores +, ==, ++ y << basado en P00. Realizar la práctica No. 18 Suma y resta de números complejos basado en P00. <p>Polimorfismo</p> <ul style="list-style-type: none"> Realizar la práctica No. 19 Cálculo de volumen de figuras geométricas basado en P00. Realizar la práctica No. 20 Creación de la clase abstracta vehículo empleando polimorfismo basado en P00. Realizar la actividad de evaluación 2.1.1 Donde: “Desarrolla programas de cómputo utilizando las propiedades de herencia y polimorfismo”. Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos. <p>Seguridad de la información</p> <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> Fabricas abstractas. Singleton o Singular. Factory Method. Prototipo. Realizar la práctica No. 21 Solicitud de usuario y contraseña basado en P00. <p>Cifrado</p> <ul style="list-style-type: none"> Realizar una investigación documental acerca de algún algoritmo de cifrado de datos y lo implementa en alguna aplicación. 	<p>Básicas:</p> <ul style="list-style-type: none"> Ceballos, F. (2015). <i>C/C++ - Curso de programación 4ta Ed.</i> Paracuellos del Jarama, Madrid, España. RA-MA. Joyanes, L. (1996). <i>Programación Orientada a Objetos.</i> Aravaca, Madrid, España. McGraw-Hill <i>Programación orientada a objetos con C++ y Java. Un intercambio interdisciplinario.</i> México: Patria. Blancarte, O. (2016). <i>Introducción a los patrones de diseño. Un enfoque práctico.</i> México. Osca Blancarte Blog. <p>Páginas web:</p> <ul style="list-style-type: none"> SoloLearn. <i>Curso de programación C++.</i> https://play.google.com/store/apps/details?id=com.sololearn.cplusplus Ruiz, R. <i>Fundamentos de Programación Orientada a Objetos.</i> Recuperado el 21/10/2017 de https://goo.gl/81KtA5 C++ Con Clase. <i>C++ Con Clase.</i> Recuperado el 21/10/2017 de http://c.conclase.net/ Microsoft. <i>Introduction to C++ and DirectX Game Development Jump Start.</i> Recuperado el 21/10/2017 de https://goo.gl/8S5dj5

- Aplicar las siguientes sentencias en el manejo de errores de excepción.
- *try*
- *catch*
- *throw*

Manejo de excepciones

- **Realizar la práctica No. 22** Manejo de errores de excepción basado en P00
- **Realizar la práctica No. 23** Manejo de excepciones en clases basado en P00.
- **Realizar la actividad de evaluación 2.2.1** Donde: *“Implementa medidas de seguridad en los programas diseñados”*.
- Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos

- Código Facilito. *Código Facilito*. Recuperado el 21/10/2017 de <http://bit.ly/CFacilito>
- Orwell DevCpp. *Dev C++*. Recuperado el 21/10/2017 de <http://bit.ly/1Xn6pkY>
- Comunidad de programadores. *La web del programador*. Recuperado el 21/10/2017 de <https://www.lawebdelprogramador.com/>
- Cueva, J. Análisis y diseño orientado a objetos. Recuperado el 21/10/2017 de <https://goo.gl/RpbU8m>
- Microsoft, *¿Qué es un Patrón de Diseño?*. Recuperado el 21/10/2017 de <https://msdn.microsoft.com/es-es/library/bb972240.aspx>
-

6 Prácticas/Actividades

Nombre del Alumno:

Unidad de Aprendizaje:

Manejo de componentes del lenguaje orientados a objetos

Resultado de Aprendizaje:

1.1 Identifica los componentes fundamentales de la programación orientada a objetos

Actividad núm.:1

Modelado de diagrama de casos de uso de una tienda de electrodomésticos.

Planteamiento:

Un negocio de venta de electrodomésticos decidió implementar y otorgar una línea de crédito a sus clientes para la compra de productos. Los créditos son solicitados por los clientes al vendedor al momento de realizar la compra y deben ser autorizados por un representante de la gerencia de créditos, y pagados por el cliente a través del débito automático en tarjetas de crédito. Si el crédito se acepta, se entrega el producto al cliente en forma inmediata.

Cada mes se cobrará de manera automática el pago de las cuotas de la tarjeta del cliente.

Se quiere modelar el proceso de solicitud, otorgamiento, y pago del crédito.

- Represente todo el proceso completo como si fuera un único caso de uso, mencionando sus pasos más importantes, sin entrar en detalles sobre alternativas.
- Identifique los distintos actores que intervienen en este proceso.
- Teniendo en cuenta su resolución del primer punto, identifique casos de uso de este proceso, que pueden ser las distintas partes del caso completo presentado en el punto a.
- A partir de los casos ya identificados, pensando en casos anteriores, siguientes, contrarios o que sean variaciones de los mismos, identifique nuevos casos de uso o alternativas entre los casos.
- Identifique casos que puedan ser extraídos de los anteriores y ser “usados” por otros casos.

Compare las ventajas y desventajas de modelar un proceso completo con un caso de uso, como en el punto a., o de modelarlo en partes separadas, como en los restantes puntos.

Nombre del Alumno:	
Unidad de Aprendizaje:	Manejo de componentes del lenguaje orientados a objetos
Resultado de Aprendizaje:	1.1 Identifica los componentes fundamentales de la programación orientada a objetos
Actividad núm.:2	Modelado de diagrama de casos de uso de una cadena de videoclubes.

Planteamiento:

La famosa cadena de videoclubes “Los Bloques de Búster” nos ha contratado con el fin de desarrollar un sistema para sistematizar sus locales.

Hasta el día de hoy se han mantenido una serie de reuniones con el cliente con el fin de determinar los requerimientos del sistema. De tales reuniones, se ha determinado lo siguiente:

El sistema deberá permitir que los clientes consulten el catálogo de películas. A partir del mismo, una vez seleccionada una película, se deberá poder acceder a la información de la misma como ser su clasificación, su género y un breve resumen de la misma. Asimismo, opcionalmente, se deberá poder consultar la disponibilidad del video.

Los empleados del videoclub deberán poder, a través del sistema, registrar las rentas y devoluciones por parte de los clientes, y consultar, dado un cliente, los videos que éste posea en renta. Si registrando una rentar, resulta que el cliente no se encuentra registrado, el sistema deberá permitir que se efectúe su alta.

Nuestro cliente también pidió que el sistema, todas las mañanas genere de forma automática un informe que muestre todos los clientes que se encuentran atrasados con sus devoluciones. Cuando se le preguntó a qué se refería con “todas las mañanas” aclaró: “Que todos los días a las 9:00 a.m. imprima o muestre por pantalla el listado de los clientes atrasados.”

Modele el enunciado anterior mediante la técnica de casos de uso.

Nombre del Alumno:	
Unidad de Aprendizaje:	Manejo de componentes del lenguaje orientados a objetos
Resultado de Aprendizaje:	1.1 Identifica los componentes fundamentales de la programación orientada a objetos
Actividad núm.:3	Modelado de diagrama de casos de uso de una máquina expendedora y de venta de bebidas.

Planteamiento:

Se ha decidido fabricar una máquina para la expedición y venta de bebidas en forma automática.

El cliente selecciona algunos de los productos ofrecidos, uno o más, por medio de la pulsación de uno o más botones. Los artículos pueden ser de distintos tipos: latas de refresco, jugos o botellas.

Solamente se puede solicitar un tipo de producto a la vez. La máquina reconoce el pedido del cliente. Si no hay en existencia le indica al cliente por medio de un mensaje.

La máquina acepta las monedas del cliente, reconociendo de distintos tipos. Si las monedas no cubren el total del importe las devuelve y le avisa al cliente por medio de un mensaje. En caso contrario, libera las bebidas solicitadas, actualiza el stock de productos e imprime un ticket.

El encargado de la reposición, repone los artículos de acuerdo a lo indicado en la pantalla (tendrá una pantalla propia, a la que accederá mediante su contraseña). Al hacerlo, debe indicarle al sistema el producto y la cantidad que se ha repuesto. Inmediatamente el sistema deberá actualizar el stock, emitir un resumen de faltante en dos copias, como constancia de reposición y factura para el poseedor de la máquina.

Modele el caso anterior mediante la técnica de casos de uso.

Nombre del Alumno:	
Unidad de Aprendizaje:	Manejo de componentes del lenguaje orientados a objetos
Resultado de Aprendizaje:	1.1 Identifica los componentes fundamentales de la programación orientada a objetos
Actividad núm.:4	Modelado de diagrama de clases del evento del Comité Olímpico Internacional.

Planteamiento:

Usted ha sido contratado por el COI (Comité Olímpico Internacional) para analizar, diseñar e implementar una solución que permita al comité tener conocimiento de todos los detalles implicados a este gran evento. Luego de reiteradas reuniones con el COI se decidió efectuar un desarrollo piloto sobre las competencias de fútbol de las Olimpiadas, a fin de determinar la efectividad del futuro sistema.

Las competencias de fútbol contarán con una serie de estadios para el desarrollo de cada uno de los partidos. Cada partido contará con la participación de dos equipos los cuales serán representación de un país invitado y un grupo designado de árbitros; cada uno de estos partidos debe proporcionarnos información acerca de: los goles marcados y las tarjetas sacadas, además de saber cuál fue la asistencia de público a cada partido y la fase de evento a la cual pertenecía. Cada equipo está conformado por 20 jugadores y un cuerpo técnico. De cada jugador queremos saber información como su nombre, fecha y lugar de nacimiento, posición que juega, etc.

Del cuerpo técnico es necesario conocer su nombre, fecha y lugar de nacimiento, cargo, etc.

De los árbitros se necesita saber su nombre, fecha y lugar de nacimiento, federación a la que pertenece y su cargo.

De los goles y las tarjetas queremos conocer el partido, el minuto y la persona que está relacionado con el gol (quién lo metió). Los estadios de las Olimpiadas son estadios que en su mayoría se han reformado o construido nuevos para darle un impulso al deporte en China. Es por ello que el comité organizador desea saber las características básicas de cada estadio como, por ejemplo, capacidad, ciudad donde están localizados, si posee techo o no, etc.

Realice el diagrama de clases en base al enunciado anterior.

Nombre del Alumno:	
Unidad de Aprendizaje:	Manejo de componentes del lenguaje orientados a objetos
Resultado de Aprendizaje:	1.1 Identifica los componentes fundamentales de la programación orientada a objetos
Actividad núm.:5	Modelado de diagrama de clases de un estacionamiento

Planteamiento:

Se desea automatizar un estacionamiento con capacidad para 400 automóviles, de acuerdo a los siguientes requisitos:

Los usuarios del estacionamiento dispondrán de una tarjeta magnética donde figura registrado su código de identificación.

A su llegada al estacionamiento, el usuario introducirá la tarjeta en el lector correspondiente, lo que hace que se eleve la barrera situada en la entrada. Esta barrera permanece levantada un cierto tiempo, descendiendo luego automáticamente.

Para salir del aparcamiento se procede de igual forma con la barrera situada a la salida.

Tanto las entradas como las salidas deben quedar registradas con objeto de realizar periódicamente una facturación a los usuarios, según el tiempo de estacionamiento consumido. Estas facturas se emitirán a petición del operador.

El sistema debe tener en cuenta la ocupación del estacionamiento, controlando un semáforo situado a la entrada. Si hay lugares disponibles libres, el semáforo debe estar verde, pasando a rojo si el estacionamiento se llena. Además, cuando el aparcamiento esté lleno no debe permitirse la entrada a nuevos vehículos.

Realice el diagrama de clases en base al enunciado anterior.

Nombre del Alumno:	
Unidad de Aprendizaje:	Manejo de componentes del lenguaje orientados a objetos
Resultado de Aprendizaje:	1.1 Identifica los componentes fundamentales de la programación orientada a objetos
Actividad núm.:6	Modelado de diagrama de secuencia de un centro de instalaciones deportivas.

Planteamiento:

Un centro de instalaciones deportivas quiere hacer una aplicación de reservas. En el centro existen instalaciones deportivas (piscinas, frontones, gimnasios y pistas de tenis). El centro en cuestión tiene socios, de los cuales se almacenan su nombre, dirección, ciudad, estado, teléfono y cuota.

Además, existen una serie de artículos que se pueden reservar si el socio lo requiere (balones, redes y raquetas).

Cada instalación es reservada por un socio en una fecha dada desde una hora de inicio hasta una hora de fin.

Cada reserva puede tener asociada uno o varios artículos deportivos que se alquilan aparte. Por ejemplo, si yo quiero hacer una reserva para jugar al tenis, tengo que reservar una instalación deportiva y si lo necesito, las raquetas.

Realice el diagrama de secuencia en base al enunciado anterior.

Nombre del Alumno:	
Unidad de Aprendizaje:	Manejo de componentes del lenguaje orientados a objetos
Resultado de Aprendizaje:	1.1 Identifica los componentes fundamentales de la programación orientada a objetos
Actividad núm.:7	Modelado de diagrama de secuencia de un controlador de pistas scalextric.

Planteamiento:

Se desea realizar un controlador para una pista de autos scalextric.

Dicho scalextric estará formado por una pista por la que circularán dos vehículos, de forma que uno será controlado por computador, y el otro por una persona.

Los vehículos tendrán asociado un identificador, una velocidad, un tiempo, un contador de vueltas, y un sensor que informe de la proximidad del vehículo contrincante.

A escasos centímetros del comienzo de una curva existirán unos sensores que informen acerca de la curvatura de la misma al vehículo.

De la misma forma, a pocos centímetros de una pendiente, también estarán colocados sensores que informen al vehículo de la inclinación de la misma.

Estos dos tipos de sensores le servirán para calcular la velocidad a tomar en el siguiente instante.

En la pista también habrá un sensor en la línea de meta para que el coche pueda llevar la cuenta de las vueltas.

El final de la carrera se determinará por el cumplimiento de una serie de vueltas configuradas a principio de la carrera, o por la salida de la pista de uno de los dos vehículos.

En ambos casos aparecerá en una pantalla el coche ganador. Si ninguno de los dos se ha salido de la pista, se incluirá el identificador de este coche junto con la duración en una tabla de récords.

Realice el diagrama de secuencia en base al enunciado anterior.

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de funciones de entrada y salida estándar basado en POO	Número:	1
Propósito de la práctica:	Realizar un programa que haga uso de funciones de entrada y salida estándar basado en POO (programación orientada a objetos). para que calcule los intereses producidos y el capital acumulado de una inversión.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evitar la manipulación de comida o líquidos cerca del equipo de cómputo - No introducir objetos extraños en las entradas físicas de dispositivos de la computadora - No utilizar imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpiar el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje <ol style="list-style-type: none"> Verifica que el equipo de cómputo se encuentra conectado. Enciende equipo de cómputo. Abre sesión de ambiente gráfico. Escribe un programa que dé como resultado los intereses producidos y el capital acumulado de una cantidad c, invertida a un interés r durante t días. <p>La fórmula utilizada para el cálculo de los intereses es:</p> <p>Siendo: I = Total de intereses producidos. c = Capital r = tasa de interés nominal en tanto por ciento. t = periodo de cálculo en días.</p> <ol style="list-style-type: none"> Ingresar al entorno de desarrollo de "C++". Transcribe el programa en el editor de "C++". Guarda el archivo que contiene el programa transcrito. Compila el programa.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>5. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.</p> <p>6. Imprime el resultado de la ejecución del programa.</p> $I = \frac{c * r * t}{360 * 100}$ <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de funciones de entrada y salida estándar basado en P00.	Número:	2
Propósito de la práctica:	Realizar un programa donde aplique funciones de entrada y salida estándar basado en POO (programación orientada a objetos), para que dé como resultado soluciones reales de ecuación de segundo grado.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje <ol style="list-style-type: none"> Verifica que el equipo de cómputo se encuentra conectado. Enciende equipo de cómputo. Abre sesión de ambiente gráfico. Escribe un programa que dé como resultado las soluciones x_1 y x_2 de una ecuación de segundo grado, de la forma: $ax^2 + bx + c = 0$ Las soluciones de una ecuación de segundo grado vienen dadas por la fórmula: $x_1 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ <p>Las soluciones son reales sólo si $b^2 - 4ac$ es mayor o igual que 0.</p> Ingresa al entorno de desarrollo de "C++". Transcribe el programa en el editor de "C++". Guarda el archivo que contiene el programa transcrito. Compila el programa.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>7. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.</p> <p>8. Imprime el resultado de la ejecución del programa.</p> <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de sentencia if-else basado en POO	Número:	3
Propósito de la práctica:	Realizar un programa que aplique la sentencia if-else basado en POO (programación orientada a objetos), para que imprima el menor de 3 números.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje <ol style="list-style-type: none"> Verifica que el equipo de cómputo se encuentra conectado. Enciende equipo de cómputo. Abre sesión de ambiente gráfico. Escribe un programa que emplee la sentencia if-else y que imprima como resultado el menor de tres números proporcionados a, b y c. Ingresa al entorno de desarrollo de "C++". Transcribe el programa en el editor de "C++". Guarda el archivo que contiene el programa transcrito. Compila el programa. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de sentencia else-if basado en POO	Número:	4
Propósito de la práctica:	Realizar un programa que aplique la sentencia else-if basado en POO (programación orientada a objetos), para que dé como resultado el importe a pagar de una compra realizada.		
Escenario:	Laboratorio de informática.	Duración	1 hora

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo. • Dispositivo de almacenamiento (USB) • Software de lenguaje de programación C++. • Impresora. 	<ul style="list-style-type: none"> • Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ∪ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribe un programa que emplee la sentencia else-if para resolver el siguiente planteamiento: <p>Al efectuar una compra en un cierto almacén, si adquirimos más de 100 unidades de un mismo artículo, nos hacen un descuento de un 40%; entre 25 y 100 un 20%; entre 10 y 24 un 10%; y no hay descuento para una adquisición de menos de 10 unidades.</p> <p>Se pide calcular el importe a pagar.</p> 5. Ingresa al entorno de desarrollo de “C++”. 6. Transcribe el programa en el editor de “C++”. 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 11. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 12. Imprime el resultado de la ejecución del programa.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de sentencia switch basado en POO	Número:	5
Propósito de la práctica:	Redactar el propósito de la práctica en la siguiente estructura: VERBO INFINITIVO + OBJETO + CONTEXTO +FINALIDAD.		
Escenario:	Proponer escenarios lo más apegado a los campos de aplicación real de la competencia, y que permitan solventar la insuficiencia de equipamiento en la realización de las mismas.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribe un programa que emplee la sentencia switch que lea una fecha representada por dos enteros, mes, y año, y que dé como resultado los días correspondientes al mes. Esto es: <p>Introducir mes (##) y año (####): 5 2009 El mes 5 del año 2009 tiene 31 días</p> <p>Hay que tener en cuenta que febrero puede tener 28 días, o bien 29 si es año bisiesto. Un año es bisiesto cuando es múltiplo de 4 y no de 100 o cuando es múltiplo de 400.</p> 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>10. Imprime el resultado de la ejecución del programa.</p> <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de sentencia <i>while</i> basado en P00	Número:	6
Propósito de la práctica:	Redactar el propósito de la práctica en la siguiente estructura: VERBO INFINITIVO + OBJETO + CONTEXTO +FINALIDAD.		
Escenario:	Proponer escenarios lo más apegado a los campos de aplicación real de la competencia, y que permitan solventar la insuficiencia de equipamiento en la realización de las mismas.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribe un programa que emplee la sentencia <i>while</i> que visualice el código ASCII de cada uno de los caracteres de una cadena de texto introducida por el teclado. 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de sentencia <i>do while</i> basado en POO	Número:	7
Propósito de la práctica:	Realizar un programa donde se use la sentencia <i>do while</i> basado en POO (programación orientada a objetos), para que visualice el código ASCII de una cadena de texto.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo. • Dispositivo de almacenamiento (USB) • Software de lenguaje de programación C++. • Impresora. 	<ul style="list-style-type: none"> • Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ∪ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribe un programa que emplee la sentencia <i>do ... while</i> que calcule la raíz cuadrada de un número <i>n</i> por el método de Newton. Este método se enuncia así: sea <i>r_i</i> la raíz cuadrada aproximada de <i>n</i>. La siguiente raíz aproximada <i>r_{i+1}</i> se calcula en función de la anterior así: $r_{i+1} = \frac{\frac{n}{r_i} + r_i}{2}$ 5. Ingresas al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de sentencia <i>for</i> basado en POO	Número:	8
Propósito de la práctica:	Realizar un programa que use sentencia <i>for</i> basado en POO (programación orientada a objetos), a fin de que imprima una cadena en forma invertida.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo. • Dispositivo de almacenamiento (USB) • Software de lenguaje de programación C++. • Impresora. 	<ul style="list-style-type: none"> • Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ∪ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribe un programa que emplee la sentencia <i>for</i> que nos pida una cadena de 10 caracteres y nos muestra dicha cadena en forma invertida. 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de arreglos	Número:	9
Propósito de la práctica:	Realizar un programa que emplee el uso de arreglos, basado en POO (programación orientada a objetos), que almacene información en un arreglo unidimensional.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribe un programa lea 10 números por teclado, los almacene en un array y muestre la media. 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

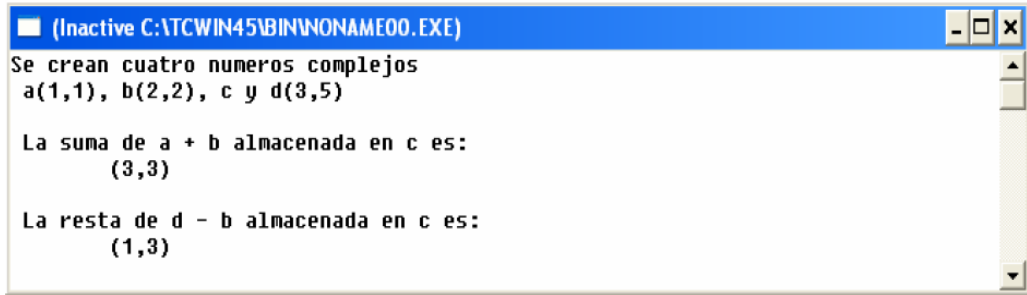
Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Uso de estructuras	Número:	10
Propósito de la práctica:	Realizar un programa que emplee el uso de estructuras, basado en POO (programación orientada a objetos), que almacene información en un arreglo que haya sido ingresada por teclado.		
Escenario:	Laboratorio de informática.	Duración	2

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 11. Verifica que el equipo de cómputo se encuentra conectado. 12. Enciende equipo de cómputo. 13. Abre sesión de ambiente gráfico. 14. Escribe un programa solicite información y esta sea capturada por teclado y sea almacenada en un arreglo y posteriormente desplegada en pantalla. 15. Ingresa al entorno de desarrollo de "C++". 16. Transcribe el programa en el editor de "C++". <pre>#include <iostream.h> using namespace std; // Una estructura simple para ver claramente su uso struct ejemplo { char pelicula[100]; int anio; }ejemplo[1]; // Función principal int main()</pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre> { // Bucle para facilitar la operación de rellenar cada una de las estructuras del array for(int j=0;j<=1;j++) { cout << " Introduce el nombre de la " << j << " pelicula " << endl; cin >> ejemplo[j].pelicula; cout << "Introduce el año de estreno de la " << j << " pelicula " << endl; cin >> ejemplo[j].anio; } // Mostramos de forma clara todos los datos obtenidos, asi se puede ver la facilidad // con la que se accede a cada uno de los campos de un array de estructuras cout << " Resultados del programa -----" << endl; cout << " Datos de la pelicula numero 1: " << endl; cout << " pelicula : " << ejemplo[1].pelicula; cout << " anio : " << ejemplo[1].anio << endl;; cout << " Datos de la pelicula numero 2: " << endl; cout << " pelicula : " << ejemplo[2].pelicula; cout << " anio : " << ejemplo[2].anio << endl; return 0; } </pre> <p>17. Guarda el archivo que contiene el programa transcrito.</p> <p>18. Compila el programa.</p> <p>19. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.</p> <p>20. Imprime el resultado de la ejecución del programa.</p> <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

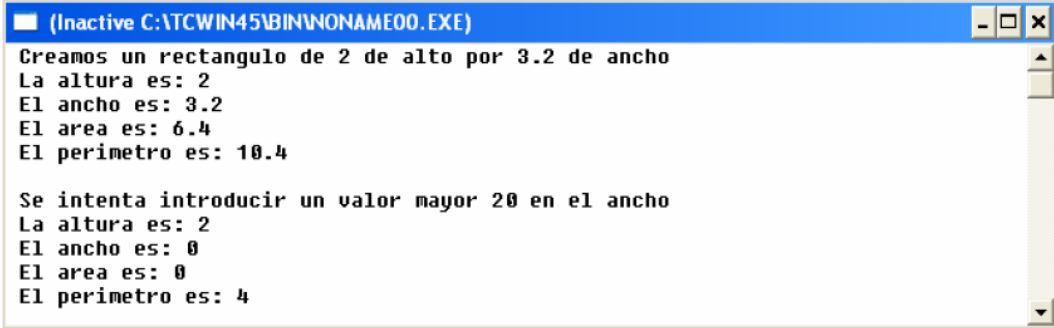
Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Implementar clases basado en P00	Número:	11
Propósito de la práctica:	Realizar un programa mediante la implementación de clases basado en P00 (programación orientada a objetos), a fin de que obtenga cálculos de datos.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo. • Dispositivo de almacenamiento (USB) • Software de lenguaje de programación C++. • Impresora. 	<ul style="list-style-type: none"> • Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ☞ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Implemente una clase llamada Complejo para realizar cálculos con números complejos sabiendo que los números complejos tienen la forma: $parteReal + parteImaginaria * i$ donde i es la raíz cuadrada de -1. <p>La clase deberá proporcionar un constructor que permita inicializar un objeto cuando se declare.</p> <p>Dicho constructor debe contener valores predeterminados. También deberá proporcionar funciones miembros de tipo <i>public</i> para cada una de las siguientes operaciones:</p> <ul style="list-style-type: none"> • Suma de dos números Complejos: las partes reales se suman y las partes imaginarias se suman. • Resta de dos números Complejos: la parte real de operando derecho se resta a la parte real del operando izquierdo, y la parte imaginaria del operando derecho se resta a la parte imaginaria del operando izquierdo. • Impresión de números Complejos de la forma (a, b), en donde a es la parte real y b es la parte imaginaria.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>NOTA: Utilice variables <i>double</i> para representar datos miembros de la clase.</p> <p>b) Escriba un método <i>main</i> cuya ejecución muestre el siguiente resultado en pantalla:</p>  <p>The screenshot shows a console window with the following text: (Inactive C:\TCWIN45\BIN\WONAME00.EXE) Se crean cuatro numeros complejos a(1,1), b(2,2), c y d(3,5) La suma de a + b almacenada en c es: (3,3) La resta de d - b almacenada en c es: (1,3)</p> <ol style="list-style-type: none"> 5. Ingresar al entorno de desarrollo de "C++". 6. Transcribir el programa en el editor de "C++". 7. Guardar el archivo que contiene el programa transcrito. 8. Compilar el programa. 9. Iniciar un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprimir el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Implementar clases basado en P00	Número:	12
Propósito de la práctica:	Realizar un programa que emplee clases, basado en P00 (programación orientada a objetos) para calcular el perímetro y el área de un rectángulo.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo. • Dispositivo de almacenamiento (USB) • Software de lenguaje de programación C++. • Impresora. 	<ul style="list-style-type: none"> • Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ☞ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Cree una clase Rectángulo con los atributos longitud y ancho, cada uno con un valor predeterminado igual a 1. <p>Proporcione funciones miembro que calculen el perímetro y el área del rectángulo. Además, proporcione las funciones <i>set</i> (establecer) y <i>get</i> (obtener) para los atributos longitud y ancho respectivamente.</p> <p>Las funciones <i>set</i> (establecer) deben verificar que longitud y ancho contengan números reales mayores que cero y menores que veinte, de lo contrario mostrarán el valor de cero.</p> <p>Escriba un método <i>main</i> cuya ejecución muestre el siguiente resultado en pantalla:</p>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	 <p>5. Ingresas al entorno de desarrollo de “C++”.</p> <p>6. Transcribes el programa en el editor de “C++”.</p> <p>7. Guarda el archivo que contiene el programa transcrito.</p> <p>8. Compila el programa.</p> <p>9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.</p> <p>10. Imprime el resultado de la ejecución del programa.</p> <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	1
Práctica:	Implementar clases basado en POO	Número:	13
Propósito de la práctica:	Programar una clase color basándose en POO (programación orientada a objetos), para que calcule el valor medio de tres componentes y ponga los tres componentes a dicho valor.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños												
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje <ol style="list-style-type: none"> Verifica que el equipo de cómputo se encuentra conectado. Enciende equipo de cómputo. Abre sesión de ambiente gráfico. En equipos electrónicos es habitual representar el color por medio de 3 componentes: rojo (R), verde (G) y azul (B). Así un color queda definido por medio de tres valores, RGB. <p>En este ejercicio a cada componente le asignaremos un valor real entre 0 y 1, de forma que algunos colores típicos quedan definidos como en la siguiente tabla:</p> <table border="1" data-bbox="871 1096 1591 1291"> <thead> <tr> <th>(R,G,B)</th> <th>COLOR</th> </tr> </thead> <tbody> <tr> <td>(1.0, 0.0, 0.0)</td> <td>Rojo</td> </tr> <tr> <td>(0.0, 1.0, 0.0)</td> <td>Verde</td> </tr> <tr> <td>(0.0, 0.0, 1.0)</td> <td>Azul</td> </tr> <tr> <td>(1.0, 1.0, 1.0)</td> <td>Blanco</td> </tr> <tr> <td>(0.0, 0.0, 0.0)</td> <td>Negro</td> </tr> </tbody> </table> <p>Se pide programar una clase "Color" que contenga:</p> <ol style="list-style-type: none"> Tres atributos privados para los componentes R, G y B. 	(R,G,B)	COLOR	(1.0, 0.0, 0.0)	Rojo	(0.0, 1.0, 0.0)	Verde	(0.0, 0.0, 1.0)	Azul	(1.0, 1.0, 1.0)	Blanco	(0.0, 0.0, 0.0)	Negro
(R,G,B)	COLOR												
(1.0, 0.0, 0.0)	Rojo												
(0.0, 1.0, 0.0)	Verde												
(0.0, 0.0, 1.0)	Azul												
(1.0, 1.0, 1.0)	Blanco												
(0.0, 0.0, 0.0)	Negro												

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>b) Un constructor predeterminado al que se le puedan pasar los valores de los tres componentes.</p> <p>El constructor utilizará un método de apoyo, de forma que si algún valor fuera negativo, se deja en 0 y si es mayor que 1, se deja en 1.</p> <p>c) Tres métodos para extraer cada componente: <i>getR()</i>, <i>getG()</i> y <i>getB()</i></p> <p>d) Un método “<i>gris()</i>” que modifica los atributos privados para pasar a <i>blanco</i> y <i>negro</i>:</p> <ul style="list-style-type: none"> - lo que se espera de este método es que calcule el valor medio de los tres componentes y ponga los tres componentes a dicho valor. <p>e) Un método “<i>filtro(cr, cg, cb)</i>” que modifica los atributos privados aplicando un filtro:</p> <ul style="list-style-type: none"> - se le pasan tres coeficientes, el de rojo, el de verde y el de azul. - el método multiplica cada componente por el coeficiente correspondiente; así <i>filtro(1.0, 0.0, 0.0)</i> sería un filtro de rojo, que sólo deja pasar el componente R. - si el resultado de multiplicar un componente por su coeficiente es mayor que 1.0, el valor debe acotarse a 1.0 <ol style="list-style-type: none"> 5. Ingresa al entorno de desarrollo de “C++”. 6. Transcribe el programa en el editor de “C++”. 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Manejo de herencia basado en POO	Número:	14
Propósito de la práctica:	Realiza un programa que maneje herencia basado en POO (programación orientada a objetos) para el cálculo de áreas de figuras geométricas.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Elabore un programa que ofrezca un menú de opciones, mediante el cual se pueda escoger calcular el área de las figuras geométricas; triángulo, cuadrado, rectángulo y círculo. <p>Una vez seleccionada la opción, que permita solicitar y leer el nombre de la figura y los datos necesarios para calcular el área correspondiente e imprimir el nombre de la figura y el área.</p> <p>Área de triángulo = $(\text{Base} * \text{Altura}) / 2$ Área de cuadrado = Lado^2 Área de círculo = $\pi * \text{radio}^2$ Área de rectángulo = $\text{Base} * \text{Altura}$</p> <p>Debe ofrecer el siguiente menú de opciones, donde está solicitando la opción deseada.</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> AREAS FIGURAS GEOMÉTRICAS </div>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños						
	<table border="1" data-bbox="873 332 1434 524"> <tr><td>1. TRIANGULO</td></tr> <tr><td>2. CUADRADO</td></tr> <tr><td>3. RECTANGULO</td></tr> <tr><td>4. CIRCULO</td></tr> <tr><td>5. FIN</td></tr> <tr><td>OPCION:</td></tr> </table> <ol style="list-style-type: none"> 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>	1. TRIANGULO	2. CUADRADO	3. RECTANGULO	4. CIRCULO	5. FIN	OPCION:
1. TRIANGULO							
2. CUADRADO							
3. RECTANGULO							
4. CIRCULO							
5. FIN							
OPCION:							

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Manejo de clases basado en POO (programación orientada a objetos)	Número:	15
Propósito de la práctica:	Realiza un programa que maneje clases basado en POO (programación orientada a objetos) para el cálculo de nómina de empleados.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo. • Dispositivo de almacenamiento (USB) • Software de lenguaje de programación C++. • Impresora. 	<ul style="list-style-type: none"> • Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ○ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. En una empresa automotriz se tienen 3 tipos de empleados: administrativos, mecánicos y vendedores. En general, para todos los empleados se tiene los datos RFC (Registro Federal de Contribuyentes), el nombre, el departamento y el puesto. En particular; para el empleado administrativo se tiene el dato sueldo mensual; para el mecánico se tiene el precio del trabajo, tantas veces como trabajos haya realizado; y para el vendedor se tiene el precio del auto, por cada auto que vendió. <p>El sueldo quincenal se calcula:</p> <ul style="list-style-type: none"> - Para el administrativo, sueldo mensual entre 2. - Para el mecánico, el 4% del valor total. - Para el vendedor, el salario mínimo, más el 2 por ciento del valor de la venta realizada.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños															
	<p>Elaborar un programa que permita procesar los empleados de la empresa e imprimir el siguiente reporte:</p> <p style="text-align: center;">Reporte de nómina quincenal</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">RFC</th> <th style="text-align: left;">Nombre</th> <th style="text-align: left;">Depto.</th> <th style="text-align: left;">Puesto</th> <th style="text-align: left;">Sueldo Quincena</th> </tr> </thead> <tbody> <tr> <td>XXXXXX</td> <td>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</td> <td>XXXXXXXXXX</td> <td>XXXXXXXXXX</td> <td>99,999.99</td> </tr> <tr> <td>XXXXXX</td> <td>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</td> <td>XXXXXXXXXX</td> <td>XXXXXXXXXX</td> <td>99,999.99</td> </tr> </tbody> </table> <p>Total 999 empleados</p> <p>La idea es que se use una superclase Empleado que contendrá los datos RFC, el nombre, el departamento, y el puesto; y los métodos para establecer y obtener cada uno de los datos. De esa superclase derivar tres subclases: EmpAdmvo, EmpMecánico y EmpVendedor; en cada una de las cuales se heredarán los datos y los métodos de la superclase; además, cada subclase de estas, deberán tener sus propios datos y métodos para establecer los datos necesarios, calcular el sueldo quincenal correspondiente y obtenerlo para imprimirlo. Asimismo, deberá haber una clase controlador que permita leer los datos y utilice el modelo para representar y solucionar el problema.</p> <ol style="list-style-type: none"> 5. Ingresa al entorno de desarrollo de “C++” . 6. Transcribe el programa en el editor de “C++” . 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>	RFC	Nombre	Depto.	Puesto	Sueldo Quincena	XXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	99,999.99	XXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	99,999.99
RFC	Nombre	Depto.	Puesto	Sueldo Quincena												
XXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	99,999.99												
XXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	99,999.99												

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Sobrecarga de Operadores + y - basado en POO	Número:	16
Propósito de la práctica:	Realizar un programa que emplee sobrecarga de operadores basado en POO (programación orientada a objetos), a fin de sumar y restar números complejos.		
Escenario:	Laboratorio de informática.	Duración	2 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribir un programa que permita realizar la suma y la resta de números complejos. Un número complejo estará definido por la clase <i>CComplejo</i> y para realizar las operaciones solicitadas esta clase incluirá un método para sobrecargar el operador + y otro para el -. 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Sobrecarga de operadores +, ==, ++ y << basado en POO	Número:	17
Propósito de la práctica:	Sobrecargar operadores aritméticos basado en POO (programación orientada a objetos), para realizar operaciones.		
Escenario:	Laboratorio de informática.	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje <ol style="list-style-type: none"> 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. <p>Dada la siguiente clase "Complejo" con las funciones miembro "suma", "resta" e "imprime":</p> <pre>#include <iostream.h> class Complejo { public: Complejo (double = 0, double = 0); Complejo suma (const Complejo &); Complejo resta (const Complejo &); void imprime(); private: double parteReal; double parteImaginaria; };</pre> <p>// IMPLEMENTACION DE LAS FUNCIONES MIEMBRO</p>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre> Complejo::Complejo (double a, double b) { parteReal = a; parteImaginaria = b; } Complejo Complejo::suma (const Complejo &x) { return Complejo (parteReal + x.parteReal, parteImaginaria + x.parteImaginaria); } Complejo Complejo::resta (const Complejo &x) { return Complejo (parteReal - x.parteReal, parteImaginaria - x.parteImaginaria); } void Complejo::imprime () { cout << "(" << parteReal << ", " << parteImaginaria << ")"; } </pre> <p>Realiza las siguientes operaciones:</p> <p>1.1</p> <ol style="list-style-type: none"> Crea la función miembro "esIguar", que devuelve un valor del tipo int, que nos dará falso o verdadero (0 o 1) dependiendo de que los dos números complejos evaluados sean distintos o iguales. Crear dos funciones miembros, que incrementen un número complejo en una unidad, tanto en la parteReal, como en la parteImaginaria, en dos versiones "preIncremento" (++X) y "posIncremento" (X++) (se incrementa el número complejo después de ser utilizado). <p>1.2</p> <ol style="list-style-type: none"> Sobrecargar el operador +, eliminando la función miembro "suma" (y opcionalmente el operador -, eliminando la función miembro "resta").

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<ul style="list-style-type: none"> b) Sobrecargar el operador ==, eliminando la función miembro "esIguar" (y opcionalmente el operador !=). c) Sobrecargar el operador ++ (como preIncremento y como posIncremento), eliminando las funciones miembro "preIncremento" y "posIncremento" (y opcionalmente el operador – en las mismas condiciones). d) Sobrecargar el operador <<, eliminando la función miembro "imprime" (y opcionalmente el operador >>). <ol style="list-style-type: none"> 4. Ingresa al entorno de desarrollo de “C++”. 5. Transcribe el programa en el editor de “C++”. 6. Guarda el archivo que contiene el programa transcrito. 7. Compila el programa. 8. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 9. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Suma y resta de números complejos basado en POO	Número:	18
Propósito de la práctica:	Sumar y restar números complejos empleando sobrecarga de operadores, basado en POO (programación orientada a objetos), para operaciones complejas.		
Escenario:	Laboratorio de informática.	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo. • Dispositivo de almacenamiento (USB) • Software de lenguaje de programación C++. • Impresora. 	<ul style="list-style-type: none"> • Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ∪ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje <ol style="list-style-type: none"> 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Genere una clase <i>CRacional</i> (fracciones) con las siguientes capacidades: <ol style="list-style-type: none"> a) Cree un constructor para prevenir un denominador 0 en una fracción, reduzca o simplifique las fracciones que no se encuentren en forma reducida y evite denominadores negativos. b) Sobrecargue los operadores de suma y resta para esta clase. (Opcional multiplicación y división) c) Sobrecargue el operador de igualdad (==) para esta clase. (Opcional (!=)) d) Sobrecargue el operador (++) como prefijo y sufijo. (Opcional (--)) e) Sobrecargue los operadores (<<) y (>>). 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.</p> <p>10. Imprime el resultado de la ejecución del programa.</p> <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Cálculo de volumen de figuras geométricas basado en POO	Número:	19
Propósito de la práctica:	Calcular volumen de figuras geométricas empleando polimorfismo, basado en POO (programación orientada a objetos), para calcular el volumen correspondiente.		
Escenario:	Laboratorio de informática.	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños			
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escribe un programa que ofrezca un menú de opciones, mediante el cual se pueda seleccionar calcular el volumen de las figuras geométricas: cubo, cilindro, cono y esfera. Una vez seleccionada la opción, que permita solicitar y leer el nombre de la figura y los datos necesarios para calcular el volumen correspondiente, imprimir el nombre de la figura y el volumen. <p>Volumen de cubo = $Arista^3$ Volumen de cilindro = $\pi r^2 h$ Volumen de cono = $1 / 3 \pi r^2 h$ Volumen de esfera = $4 / 3 \pi r^3$</p> <p>Debe ofrecer el siguiente menú de opciones para solicitar la opción deseada.</p> <table border="1" data-bbox="1079 1308 1677 1403"> <tr> <td>VOLUMENES FIGURAS GEOMÉTRICAS</td> </tr> <tr> <td>1. CUBO</td> </tr> <tr> <td>2. CILINDRO</td> </tr> </table>	VOLUMENES FIGURAS GEOMÉTRICAS	1. CUBO	2. CILINDRO
VOLUMENES FIGURAS GEOMÉTRICAS				
1. CUBO				
2. CILINDRO				

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños			
	<table border="1" data-bbox="1079 334 1677 431"> <tr> <td>3. CONO</td> </tr> <tr> <td>4. ESFERA</td> </tr> <tr> <td>OPCION:</td> </tr> </table> <p data-bbox="869 464 1923 764">La idea es que se use una superclase abstracta <i>Figura</i> que contendrá el dato nombre y los métodos para establecerlo y obtenerlo; además el dato volumen, un método abstracto para calcular el volumen y un método para obtenerlo e imprimirlo. De esa superclase derivar cuatro subclases: Cubo, Cilindro, Cono y Esfera; en cada una de las cuales se heredarán los datos y los métodos de la superclase <i>Figura</i>. Cada subclase de éstas, deberá tener sus propios datos y los métodos necesarios para establecerlos; además del método calcular el volumen de la figura correspondiente. En virtud de que calcular volumen es un método abstracto heredado de la superficie abstracta <i>Figura</i>, cada una de las subclases derivadas, lo deberá implementar de acuerdo con la forma que le corresponda; aplicando el polimorfismo.</p> <ol data-bbox="823 800 1898 1011" style="list-style-type: none"> Ingresar al entorno de desarrollo de "C++". Transcribir el programa en el editor de "C++". Guardar el archivo que contiene el programa transcrito. Compilar el programa. Iniciar un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. Imprimir el resultado de la ejecución del programa. <p data-bbox="823 1029 1923 1084">Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p data-bbox="823 1105 1923 1161">Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>	3. CONO	4. ESFERA	OPCION:
3. CONO				
4. ESFERA				
OPCION:				

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Creación de clase abstracta vehículo empleando polimorfismo basado en POO	Número:	20
Propósito de la práctica:	Crear clase abstracta vehículo empleando polimorfismo, basado en POO (programación orientada a objetos), que obtenga clases de vehículos, atributos como costo y año del vehículo.		
Escenario:	Laboratorio de informática.	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Crear una clase base abstracta <i>vehículo</i> que contenga una función virtual para mostrar los atributos de un objeto en pantalla y dos clases derivadas que deberán concretar la clase anterior: vehículo Terrestre y vehículo Aéreo. La clase base definirá dos atributos: costo y año del vehículo y dos funciones para obtener sus valores. Las clases derivadas vehículo Terrestre y vehículo Aéreo contienen los atributos kilometraje y horas de vuelo respectivamente. Además, deberán ofrecer funciones para obtener y establecer esos atributos. Escribir una función de prueba main() que cree objetos de las clases derivadas y un apuntador de clase base que haga uso de la función virtual que se ha concretado en cada una de las clases derivadas. 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>10. Imprime el resultado de la ejecución del programa.</p> <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Solicitud de usuario y contraseña basado en POO	Número:	20
Propósito de la práctica:	Solicitar en un programa de cómputo usuario y contraseña, empleando el patrón de diseño singleton, basado en POO (programación orientada a objetos), para restringir el acceso a las aplicaciones.		
Escenario:	Laboratorio de informática.	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> Evita la manipulación de comida o líquidos cerca del equipo de cómputo No introduce objetos extraños en las entradas físicas de dispositivos de la computadora No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Realizar una aplicación empleando el patrón de diseño Singleton, que solicite al usuario su nombre y una contraseña. <p>Si la autenticación es correcta mostrará una ventana en la que aparecerá información sobre quién y cuándo ha ejecutado la aplicación, en caso contrario terminar la aplicación</p> <p>El diagrama de clases del patrón Singleton es el siguiente:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">Singleton</p> <hr/> <ul style="list-style-type: none"> - <u>singleton</u> : Singleton - Singleton() + <u>getInstance()</u> : Singleton </div> <pre> template <class T> class Singleton { </pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre> public: static T& Instance() { static T laInstanciaSingleton; //asumir T posee un constructor por defecto return laInstanciaSingleton; } }; class Singleton : public Singleton<Singleton> { friend class Singleton<Singleton>; //para dar acceso al constructor privado de SoloUno //..definir aquí el resto de la interfaz }; </pre> <ol style="list-style-type: none"> 5. Ingresa al entorno de desarrollo de “C++”. 6. Transcribe el programa en el editor de “C++”. 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa. 9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido. 10. Imprime el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Manejo de errores de excepción basado en POO	Número:	22
Propósito de la práctica:	Modificar el código propuesto basado en POO (programación orientada a objetos), para capturar errores de excepción.		
Escenario:	Laboratorio de informática.	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> - Evita la manipulación de comida o líquidos cerca del equipo de cómputo - No introduce objetos extraños en las entradas físicas de dispositivos de la computadora - No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora - Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Modifica el siguiente programa, de tal forma que haga uso del manejo de excepciones para controlar los posibles errores que se puedan dar: <pre>#include <iostream> using namespace std; int main() { int *x = 0; int y = 100000000; x = new int [y]; if (x) { x[10] = 0; cout << "punter: " << x << endl; delete [] x; } }</pre>

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<pre>else { cout << "Memoria insuficiente" << endl; } cin.get(); return 0; }</pre> <ol style="list-style-type: none">5. Ingresar al entorno de desarrollo de "C++".6. Transcribir el programa en el editor de "C++".7. Guardar el archivo que contiene el programa transcrito.8. Compilar el programa.9. Iniciar un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.10. Imprimir el resultado de la ejecución del programa. <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

Unidad de aprendizaje:	Manejo de componentes del lenguaje orientados a objetos	Número:	2
Práctica:	Manejo de excepciones en clases basado en POO	Número:	23
Propósito de la práctica:	Realizar programa de excepciones en clases basado en POO (programación orientada a objetos), para controlar el manejo de errores en aplicaciones.		
Escenario:	Laboratorio de informática.	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo. Dispositivo de almacenamiento (USB) Software de lenguaje de programación C++. Impresora. 	<ul style="list-style-type: none"> Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: <ul style="list-style-type: none"> Evita la manipulación de comida o líquidos cerca del equipo de cómputo No introduce objetos extraños en las entradas físicas de dispositivos de la computadora No utiliza imanes cerca de discos compactos, memorias extraíbles o de la computadora Limpia el área de trabajo, prepara herramientas y los materiales a utilizar Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envío a reciclaje 1. Verifica que el equipo de cómputo se encuentra conectado. 2. Enciende equipo de cómputo. 3. Abre sesión de ambiente gráfico. 4. Escriba un programa que declare una clase llamada Leerdatos. Esta clase debe tener como miembros públicos 4 funciones: <pre>static void LeerInt (int &); static void LeerString (char *); static void LeerDouble (double &); static void LeerChar (char &);</pre> <p>Se deberán definir estas funciones, de tal forma que se almacene la variable que se pasa como parámetro un valor pedido por teclado según el tipo. Observe que los parámetros son referencias. Se debe hacer uso del manejo de excepciones para controlar errores que se puedan producir durante la lectura. A continuación escriba una función main() que haga uso de dicha clase y que ponga a prueba el código que maneja las posibles excepciones.</p> 5. Ingresa al entorno de desarrollo de "C++". 6. Transcribe el programa en el editor de "C++". 7. Guarda el archivo que contiene el programa transcrito. 8. Compila el programa.

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
	<p>9. Inicia un proceso de depuración del programa (corrección de errores) hasta cerciorarse que el programa cumpla con el objetivo establecido.</p> <p>10. Imprime el resultado de la ejecución del programa.</p> <p>Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.</p> <p>Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.</p>

II. Guía de evaluación del módulo Programación orientada a objetos

7 Descripción

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

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.

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 de que se trate, deberá **ir acumulando** dichos porcentajes a lo largo del período para estar en condiciones de acreditar el mismo. Cada una de estas ponderaciones dependerá de la relevancia que tenga 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 asimismo establecida en la **Tabla de ponderación**, la cual está desarrollada en una hoja de cálculo que permite, tanto al alumno como al docente, ir observando y calculando los avances en términos de porcentaje, que se van alcanzando. Esta tabla de ponderación contiene los Resultados de Aprendizaje y las Unidades a las cuales pertenecen. Asimismo, indica, en la columna de actividades de evaluación, la codificación asignada a ésta desde el programa de estudios y que a su vez queda vinculada al Sistema de Evaluación Escolar SAE. Las columnas de aspectos a evaluar, corresponden al tipo de aprendizaje que se evalúa: **C = conceptual; P = Procedimental y A = Actitudinal**. Las siguientes tres columnas indican, en términos de porcentaje: la primera el **peso específico** asignado desde el programa de estudios para esa actividad; la segunda, **peso logrado**, es el nivel que el alumno alcanzó con base en las evidencias o desempeños demostrados; la tercera, **peso acumulado**, se refiere a la suma de los porcentajes alcanzados en las diversas actividades de evaluación y que deberá acumular a lo largo del ciclo escolar.

Otro elemento que complementa a la matriz de ponderación es la **rúbrica o matriz de valoración**, que establece los **indicadores y criterios** a considerar para evaluar, ya sea un producto, un desempeño o una actitud. Una matriz de valoración o rúbrica es, como su nombre lo indica, una matriz de doble entrada en la cual se establecen, por un lado, los **indicadores** o aspectos específicos que se deben tomar en cuenta como **mínimo indispensable** para evaluar si se ha logrado el resultado de aprendizaje esperado y, por otro, los criterios o **niveles de calidad o satisfacción alcanzados**. En las celdas centrales se describen los criterios que se van a utilizar para evaluar esos indicadores, explicando cuáles son las características de cada uno. Los criterios que se han establecido son: **Excelente**, en el cual, además de cumplir con los estándares o requisitos establecidos como necesarios en el logro del producto o desempeño, es propositivo, demuestra iniciativa y creatividad, o que va más allá de lo que se le solicita como mínimo, aportando elementos adicionales en pro del indicador; **Suficiente**, si cumple con los estándares o requisitos establecidos como necesarios para demostrar que se ha desempeñado adecuadamente en la actividad o elaboración del producto. Es en este nivel en el que podemos decir que se ha adquirido la competencia. **Insuficiente**, para cuando no cumple con los estándares o requisitos mínimos establecidos para el desempeño o producto.

8 Tabla de ponderación

UNIDAD	RA	ACTIVIDAD DE EVALUACIÓN	ASPECTOS A EVALUAR			% Peso Específico	% Peso Logrado	% Peso Acumulado
			C	P	A			
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 ya sea para la autenticación como para el cifrado de los datos.	2.2.1	▲	▲	▲	15	
% PESO PARA LA UNIDAD						50		
PESO TOTAL DEL MÓDULO						100		

**9 Desarrollo de actividades de
evaluación.**

10 Matriz de valoración o rúbrica

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	PROO 03	Nombre del módulo:	Programación orientada a objetos	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	1.1. Diseña modelos bajo el enfoque de la metodología orientada a objetos.		Actividad de evaluación:	1.1.1. Diseña modelos bajo el enfoque de la programación orientada a objetos, que contenga.	

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
ELEMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS 7.3	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 	<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. 	Falta alguna característica: <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.

		los diferentes elementos de la Programación Orientada a Objetos.		
LENGUAJE DE MODELADO 7.3	30	<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. • Presenta en orden los diagramas elaborados. <p>Emplea una herramienta de modelado UML para diseñar los diagramas y los presenta en medio magnético.</p>
SOFTWARE DE DESARROLLO 5.2	20	<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 	<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, <p>Presenta un análisis de ventajas de portabilidad entre las diferentes plataformas que presentan algunos lenguajes de programación orientados a objetos.</p>

		objetos.		
PRESENTACION DE RESULTADOS 7.3	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. • Redacta documento aplicando las reglas ortográficas y gramaticales. 	<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 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>
ACTITUDES 8.3	10	<ul style="list-style-type: none"> • Cumple con asistencia total y tiene 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. • Trabaja con limpieza y orden. • 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. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo 	<p>Omite alguna de las siguientes características:</p> <ul style="list-style-type: none"> • Cumple con asistencia total y tiene 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. • Trabaja con limpieza y orden. <p>Tiene disposición y asume rol asignado en el trabajo colaborativo</p>
	100			

Siglema:	PROO 03	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 7.3	15	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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. 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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. 	<p>Omite alguna característica al emplear el lenguaje de programación especificado por el docente, cuando:</p> <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. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
SENTENCIAS DE CONTROL 5.2	15	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> • Desarrolla programas utilizando sentencias de control: <ul style="list-style-type: none"> - If else - else if 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> • Desarrolla programas utilizando sentencias de control: <ul style="list-style-type: none"> - If else - else if 	<p>Omite alguna característica al emplear lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> • Desarrolla programas utilizando sentencias de control: <ul style="list-style-type: none"> - If else

		<ul style="list-style-type: none"> - 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 <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. 	<ul style="list-style-type: none"> - 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. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
ARREGLOS Y ESTRUCTURAS 7.3	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 5.2</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>PRESENTACION DE RESULTADOS 5.2 y 7.3</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. 	<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. 	<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,) • Cumple con los criterios de contenido y presentación establecidos.

		<ul style="list-style-type: none"> • 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. • Incluye referencias documentales y/o electrónicas empleadas. <p>Presenta el documento impreso limpio, respetando la estructura definida para su diseño.</p>
ACTITUDES 8.3	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. • Trabaja con limpieza y orden. <p>Tiene disposición y asume rol asignado en el trabajo colaborativo</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 • Trabaja con limpieza y orden. • 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. • Trabaja con limpieza y orden. <p>Tiene disposición y asume rol asignado en el trabajo colaborativo</p>
	100			

Siglema:	PROO 03	Nombre del módulo:	Programación orientada a objetos	Nombre del alumno:	
Docente evaluador:		Grupo:		Fecha:	
Resultado de aprendizaje:	2.1. Elabora programas de cómputo 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 7.3	30	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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. 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <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. 	<p>Omite dos o más características al emplear lenguaje de programación especificado por el docente, cuando:</p> <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. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
SOBRECARGA DE OPERADORES 5.4	20	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> • Desarrolla el código de programación en el que utilice la sobrecarga de operadores. 	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> • Desarrolla el código de programación en el que utilice la sobrecarga de operadores. 	<p>Omite dos o más características al emplear lenguaje de programación especificado por el docente, cuando:</p>

		<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. • 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. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
<p>POLIMORFISMO 7.3</p>	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>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
<p>PRESENTACION DE RESULTADOS 6.1</p>	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 	<ul style="list-style-type: none"> • Presenta la información descrita en los apartados anteriores en formato impreso y digital. 	<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

		<p>texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,)</p> <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. <p>Presenta el documento impreso limpio, respetando la estructura definida para su diseño.</p>	<ul style="list-style-type: none"> • 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>impreso y digital (elaborados en un procesador de texto y/o presentador gráfico), con estructura (caratula, objetivo y contenido,)</p> <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. <p>Presenta el documento impreso limpio, respetando la estructura definida para su diseño.</p>
<p>ACTITUDES 7.1</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. • Trabaja con limpieza y orden. • 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 • Trabaja con limpieza y orden. • 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. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo
	<p>100</p>			

Siglema:	PROO 03	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 ya sea para la autenticación como para el cifrado de los datos.		Actividad de evaluación:	2.2.1. Implementa medidas de seguridad en los programas diseñados haciendo uso de patrones de diseño	

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
SEGURIDAD DE LA INFORMACIÓN 5.4	30	<p>Emplea lenguaje de programación especificado por el docente, cuando:</p> <ul style="list-style-type: none"> • Elabora investigación documental acerca de distintos patrones de diseño existentes para implementar la autenticación en aplicaciones orientadas a objetos. 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. 	<p>Emplea lenguaje de programación especificado por el docente, cuando</p> <ul style="list-style-type: none"> • Elabora investigación documental acerca de distintos patrones de diseño existentes para implementar la autenticación en aplicaciones orientadas a objetos. 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. 	<p>Carece de alguna o varias de las siguientes características:</p> <ul style="list-style-type: none"> • Elabora investigación documental acerca de distintos patrones de diseño existentes para implementar la autenticación en aplicaciones orientadas a objetos. 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.

		<ul style="list-style-type: none"> • Presenta el código fuente en orden y sintaxis correcta del patrón de diseño implementado. • Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada. 		<ul style="list-style-type: none"> • Presenta el código fuente en orden y sintaxis correcta del patrón de diseño implementado. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
CIFRADO 5.4	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 7.3	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. <p>Inserta comentarios en el código</p>	<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.

		<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. 	desarrollado para facilitar la comprensión y documentación de la programación realizada.	<ul style="list-style-type: none"> • Presenta el código fuente en orden y sintaxis correcta del manejo de excepciones. <p>Inserta comentarios en el código desarrollado para facilitar la comprensión y documentación de la programación realizada.</p>
PRESENTACION DE RESULTADOS 6.1	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. <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>Presenta el documento impreso limpio, respetando la estructura definida para su diseño.</p>
ACTITUDES 7.1 y 7.3	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 	<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 • Trabaja con limpieza y orden. • 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. • Trabaja con limpieza y orden.

		fecha previa a la establecida por el docente. • Trabaja con limpieza y orden. Tiene disposición y asume rol asignado en el trabajo colaborativo		• Muestra disposición y asume rol asignado en el trabajo colaborativo
	100			