



### Introducción a la unidad

La programación orientada a objetos (POO) surge de la necesidad de contar con lenguajes que pudieran implementar soluciones computables de una forma más parecida a la que utilizamos en la vida real, ya que nosotros concebimos nuestro mundo como un conjunto de cosas u objetos con propiedades o atributos que los definen como tal.

### Objetivo particular de la unidad

Explicar los conceptos más importantes que dan soporte y fundamento al paradigma de programación orientado a objetos (clase, objeto, herencia, polimorfismo y encapsulamiento), y aplicarlos en la resolución de problemas algorítmicos propios de la automatización de la información y manipulación de datos.

### LO QUE SÉ

Antes de iniciar con el estudio de la unidad, escribe tu propia definición sobre las siguientes palabras.

- Clase
- Objeto
- Programación Orientada a objetos
- Polimorfismo

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**.



## Unidad III. Paradigma orientado a objetos



### Temas de la unidad III

1. Definición
2. Clases y objetos
3. Atributos y métodos
4. Relaciones estáticas y dinámicas entre objetos
5. Herencia
6. Polimorfismo
  - 6.1. Sobrecarga
  - 6.2. Sobre-escritura
7. Encapsulamiento (ocultamiento de la información)
8. Diagrama de clases
9. Campos de aplicación

### Resumen de la unidad

Actualmente una de las áreas de mayor aplicación tanto en la industria como en el ámbito académico para la ingeniería de software es la Programación Orientación a Objetos (POO). La base conceptual son los objetos, sus relaciones e interrelación.

Los objetos son entidades que tienen:

- Un estado: los datos que un objeto tiene en un momento determinado.
- Un comportamiento: los procedimientos o métodos que contiene un objeto, es decir, las operaciones asociadas al objeto.
- Una identidad: un identificador que hace una referencia única al objeto.

Bajo esta visión, un programa viene a ser representado por un conjunto de objetos que colaboran entre sí para realizar una o más tareas.



## Unidad III. Paradigma orientado a objetos



### Tema 1. Definición

#### Objetivo del tema

Identificar el concepto de programación orientada a objetos.

#### Desarrollo

Es el paradigma que define objetos y clases como la base para la programación. Cada objeto está definido por sus atributos y su comportamiento está definido por las operaciones que dichos objetos pueden hacer. La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.<sup>1</sup> Programación orientada a objetos” 01/08, material en línea, disponible en: <http://ceaelmrabet.blogspot.com/2008/01/oop.html>, recuperado el 28/01/09. Su uso se popularizó en los 90. Actualmente son muchos los lenguajes de programación que soportan la orientación a objetos. Ejemplos: Java, C++, Smalltalk, PHP y Ruby.

---

<sup>1</sup> Programación orientada a objetos” 01/08, material en línea, disponible en: <http://ceaelmrabet.blogspot.com/2008/01/oop.html>, recuperado el 28/01/09.



## Unidad III. Paradigma orientado a objetos



### ACTIVIDAD 1

Revisa el contenido las páginas de Programación orientada a objetos tanto de Wikipedia [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_orientada\\_a\\_objetos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) como de Monografias.com

<http://www.monografias.com/trabajos/objetos/objetos.shtml>

Elabora una tabla de comparación de ambas páginas sobre los enfoques que presentan sobre el paradigma orientado a objetos.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

### Bibliografía básica

Autor	Capítulo	Páginas

### Sitios electrónicos

Sitio	Descripción



## Unidad III. Paradigma orientado a objetos



### Tema 2. Clases y objetos

#### Objetivos del tema

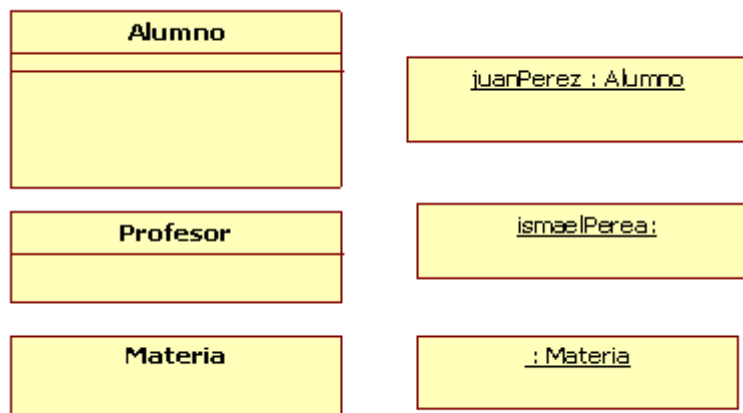
Identificar la relación existente entre la definición de Clase y Objeto.

#### Desarrollo

Una clase es una plantilla que encapsula los datos y las abstracciones de datos necesarios para describir el contenido y comportamiento de alguna entidad del mundo real. Es una descripción generalizada (patrón o plantilla) que describe una colección de objetos similares.

Un objeto es una instancia de una clase específica: los objetos heredan los atributos y operaciones de una clase (su clase padre). Todos los objetos de una clase tienen el mismo conjunto de atributos y el mismo número de operaciones. Difieren solamente en los valores de sus atributos respectivos.

Para modelar una clase y un objeto se utiliza un lenguaje de modelado llamado UML (Lenguaje Unificado de Modelado):



#### Tokens o símbolos de UML que se utilizan para modelar clases y objetos



## Unidad III. Paradigma orientado a objetos



### ACTIVIDAD 1

Realizar la lectura de Hora 2 – Orientación a objetos del libro ‘Aprendiendo UML en 24 Horas’. Elabora posteriormente una lista de clases y objetos que puedas identificar en tu entorno (laboral, académico y/o doméstico).

#### Horas **(ANEXO 1)**

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



### Unidad III. Paradigma orientado a objetos



#### Autoevaluación

Identifica de la lista que se presenta a continuación, aquellos elementos que corresponden a las clases y aquellas que representan objetos. Después arrastra cada oración al lugar correspondiente. Al final obtendrás tu calificación de manera automática.

Clase	Objeto

- El nuevo modelo de carro
- El carro de mi papá
- Las coníferas del país
- El oso panda del zoológico del bosque de Chapultepec
- Los alumnos de la Facultad de Contaduría y Administración
- Los clientes de 'La Farmacia de Dios'
- La factura nº 28934 de la tienda 'El amigo'
- La cuenta de la mesa 4
- Las cuentas del restaurant 'La Oaxaqueña'
- El oso polar
- Auto de carrera
- El estacionamiento de la fábrica
- El alumno con el número de cuenta 446682271
- Cadena de tiendas 'El paisano'



### Unidad III. Paradigma orientado a objetos



#### Bibliografía básica

Autor	Capítulo	Páginas

#### Sitios electrónicos

Sitio	Descripción







## Unidad III. Paradigma orientado a objetos



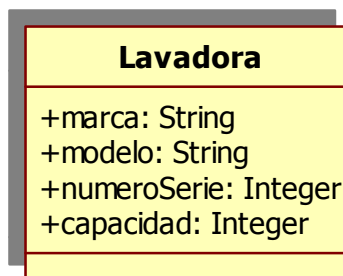
### Tema 3. Atributos y métodos

#### Objetivos del tema

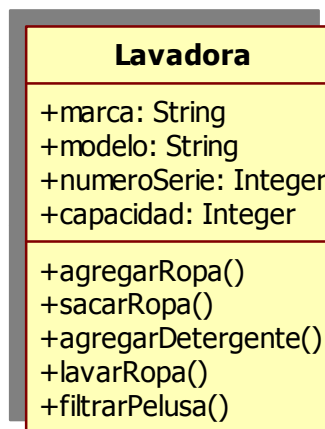
Identificar la definición de Clase y su relación con los conceptos Atributos y Métodos, para la descripción de propiedades y comportamiento de los objetos que se especifican.

#### Desarrollo

Un **atributo** es una propiedad, rasgo o característica de una clase y describe un rango de valores que la propiedad podrá contener en los objetos (instancias).



Las **operaciones** también llamadas **métodos o servicios** son algo que la clase puede hacer o que nosotros u otra clase pueden hacer a una clase.





## Unidad III. Paradigma orientado a objetos



La **responsabilidad** es una descripción de lo que hará la clase, es decir, de lo que sus atributos y operaciones intentan realizar en conjunto.

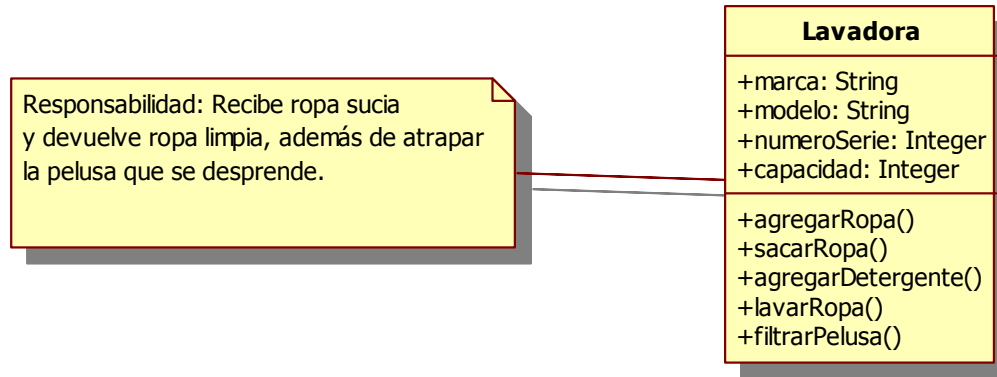


Figura 3.2. Ejemplo de responsabilidad

### ACTIVIDAD 1

Realiza la lectura de Hora3 – Uso de la orientación a objetos del libro ‘Aprendiendo UML en 24 Horas’ (**ANEXO 1**). Elabora un resumen de los puntos relevantes no mayor a una cuartilla contesta las siguientes preguntas:

- ¿Qué es un atributo?
- ¿Qué es un método?
- ¿A qué se denomina comportamiento?
- ¿Cómo distinguir atributos y métodos de una clase de objetos?

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



## Unidad III. Paradigma orientado a objetos



### Autoevaluación

Dada la siguiente clase:

Señala con color amarillo los datos y con color azul los métodos.

Para enviar su actividad a su profesor, pulse el botón Guardar imagen, el ejercicio se salvará como archivo imagen; presione **Examinar** para localizarlo en su computadora y por último **Subir este archivo**.

Vehiculo
dueno : string puertas : int ruedas : int
Vehiculo() acelerar()

### Bibliografía básica

Autor	Capítulo	Páginas

### Sitios electrónicos

Sitio	Descripción



## Unidad III. Paradigma orientado a objetos



### Tema 4. Relaciones estáticas y dinámicas entre objetos

#### Objetivo del tema

Identificar las relaciones que pueden representarse en el diseño de objetos.

#### Desarrollo

Entre clases se pueden entablar relaciones, ya que muy difícilmente una clase está aislada del resto. Una relación es una conexión entre elementos. Las tres relaciones más importantes son:

- Dependencia
- Generalización (herencia)
- Asociación:
  - Agregación
  - Composición

Una **dependencia** es una relación de uso que declara que un cambio en la especificación de un elemento afecta a otro elemento que la utiliza, pero no necesariamente a la inversa.



Una **asociación** es una relación estructural que especifica que los objetos de un elemento están conectados con los objetos de otro.

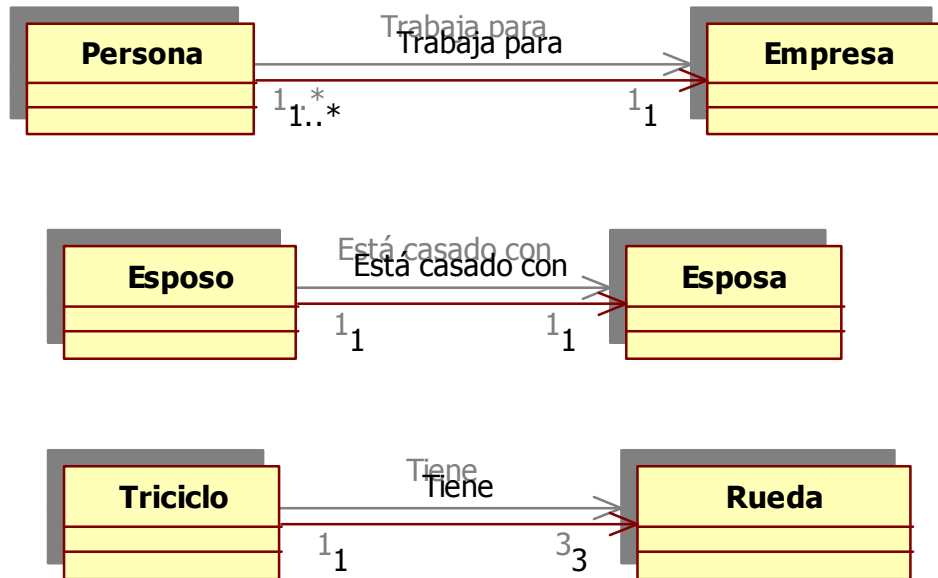




## Unidad III. Paradigma orientado a objetos



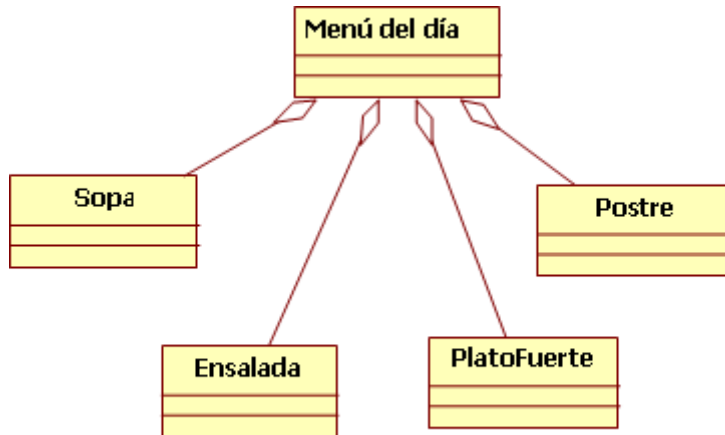
**Multiplicidad:** Como ya se dijo, una asociación representa una relación estructural entre objetos. En algunas ocasiones es necesario señalar cuántos objetos de una clase pueden relacionarse con un objeto de la clase asociada. El “cuántos” se conoce como multiplicidad.



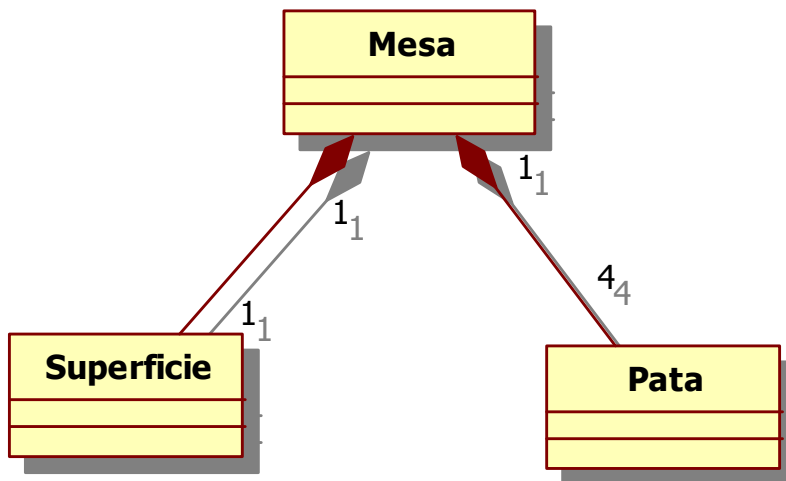
**Agregación:** A veces se desea modelar una relación “todo-parte”, en la cual una clase representa una cosa grande (el “todo”), que consta de elementos más pequeños (las “partes”). Dicho de otra forma, una clase consta de otras clases. Este tipo de asociación se llama agregación.



## Unidad III. Paradigma orientado a objetos



**Composición:** Una composición es un tipo especial de agregación donde cada componente dentro de una composición puede pertenecer tan sólo a un todo.





## Unidad III. Paradigma orientado a objetos



### ACTIVIDAD 1

Realizar la lectura de Hora 5 – Agregación, composición y realización del libro ‘Aprendiendo UML en 24 Horas’ (**ANEXO 1**). Elabora un resumen de los puntos relevantes y contesta las siguientes preguntas:

- ¿Cómo se indica la agregación entre clases?
- ¿Qué criterios se siguen para identificar cuando es aplicable la agregación?
- ¿Qué indica multiplicidad en una relación de clases?
- ¿Las asociaciones determinan la complejidad de los objetos que se modelan?

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

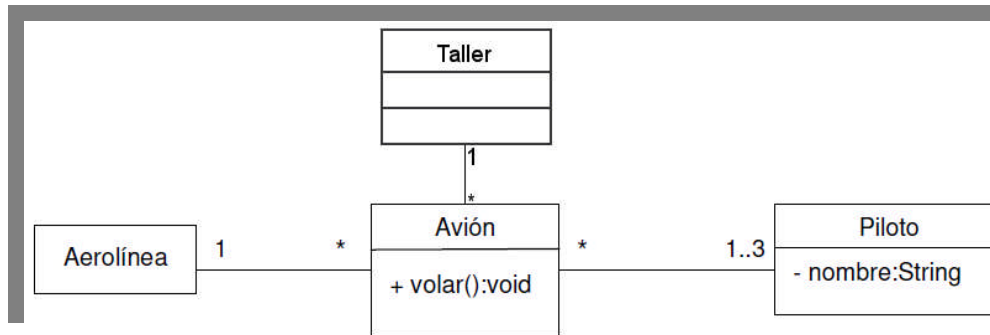


## Unidad III. Paradigma orientado a objetos



### Autoevaluación

Sean las siguientes clases relacionadas.



Elige la frase que complete correctamente cada oración. Al final obtendrás tu calificación de manera automática.

Indicar las relaciones que se indican a continuación:

1. Un Piloto \_\_\_\_\_ Aviones
2. Un Avión \_\_\_\_\_ Pilotos.
3. Una Aerolínea \_\_\_\_\_ Aviones.
4. Un Avión \_\_\_\_\_ Aerolínea
5. Un Taller \_\_\_\_\_ Aviones

### Bibliografía básica

Autor	Capítulo	Páginas

### Sitios electrónicos

Sitio	Descripción





## Unidad III. Paradigma orientado a objetos



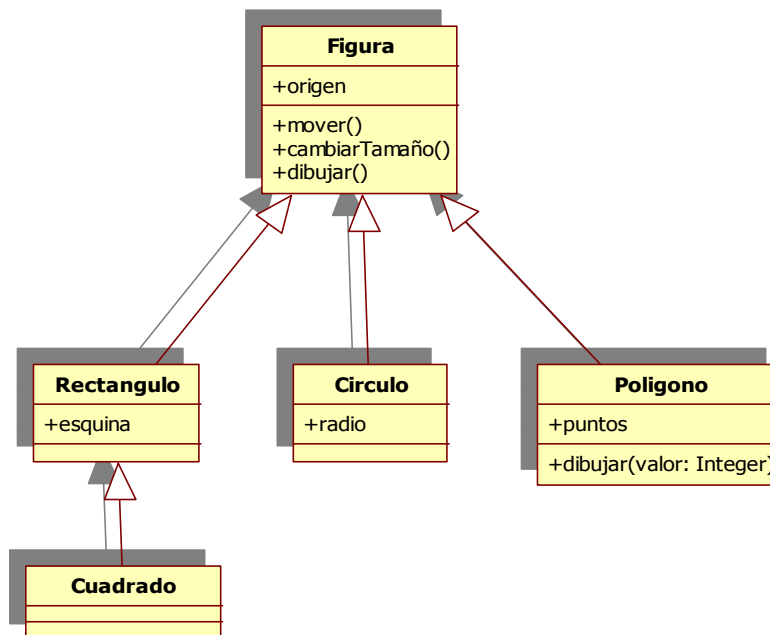
### Tema 5. Herencia (generalización)

#### Objetivo del tema

Identificar la utilidad del concepto herencia y su aplicación en el diseño de clases.

#### Desarrollo

Una **generalización** es una relación entre un elemento general (llamado superclase o clase padre) y un caso más específico (especialización) de ese elemento (llamado subclase o clase hija). La generalización es la implementación del concepto de **herencia**. En la generalización un hijo puede sustituir al padre. La clase hija hereda atributos y operaciones de su o sus padres. Una clase hija puede tener sus propios atributos y operaciones e incluso redefinir o implementar dichas operaciones a través del concepto de polimorfismo.

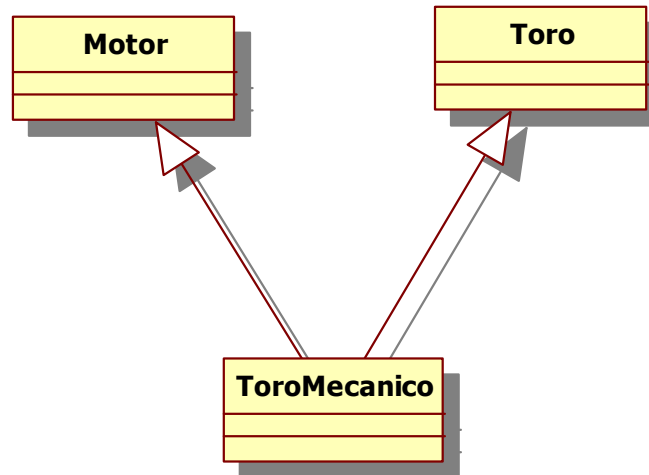




## Unidad III. Paradigma orientado a objetos



**Herencia múltiple:** Una clase puede ser hija de dos o más clases. A esto se le llama herencia múltiple.



### ACTIVIDAD 1

Revisa la definición de Herencia de Hora 4 –Uso de relaciones del libro ‘Aprendiendo UML en 24 Horas’ (**ANEXO 1**). Indica los tipos de herencia que pueden encontrarse implementadas en los lenguajes orientados a objetos (simple o múltiple) y su significado.

Elabora un cuadro con los datos obtenidos donde indiques los tipos de herencia y su significado.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

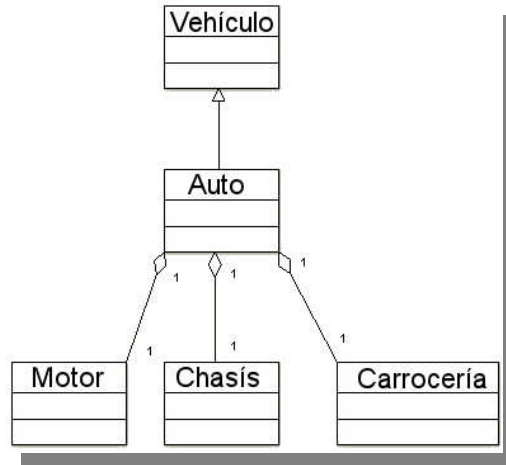


## Unidad III. Paradigma orientado a objetos



### Autoevaluación

Dado el siguiente diagrama de clases



Arrastra a los espacios en blanco, las frases que completen correctamente cada oración.

1. Un Auto \_\_\_\_\_ Vehículo
2. Un Motor \_\_\_\_\_ Auto
3. Un Auto \_\_\_\_\_ Chasis
4. Un auto \_\_\_\_\_ un Motor, un Chasis y una Carrocería
5. Una Carrocería \_\_\_\_\_ Auto

### Bibliografía básica

Autor	Capítulo	Páginas

### Sitios electrónicos

Sitio	Descripción



## Unidad III. Paradigma orientado a objetos



### Tema 6. Polimorfismo

#### Objetivo del tema

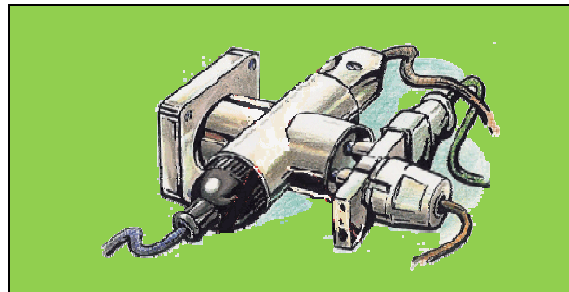
Reconocer los conceptos Sobrecarga y Sobre-escritura.

#### Desarrollo

Es cuando las operaciones de los objetos de una misma clase tienen comportamientos diferentes. Incluso el comportamiento de la clase padre difiere del comportamiento de las clases hijas. Es obvio que comparten el mismo nombre. Existen dos tipos de polimorfismo:

#### Sobrecarga

Es cuando un método (operación) de una clase recibe diferentes parámetros para realizar su tarea.



#### Sobreescritura

Es cuando el comportamiento de un método (operación) es diferente con respecto a las demás instancias o incluso al mismo padre. Esto significa que se redefine el código del método. Cuando esto pasa, es necesario definir al método padre como abstracto; esto es, no definir ningún comportamiento en el padre, sólo el nombre, para dejarlo a las instancias que lo definan según sus necesidades.



## Unidad III. Paradigma orientado a objetos



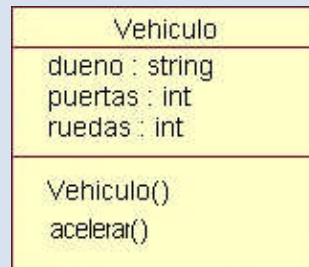
### ACTIVIDAD 1

Con base en la lectura de las páginas Sobrecarga de métodos y de constructores [http://profesores.fi-b.unam.mx/carlos/java/java\\_basico4\\_6.html](http://profesores.fi-b.unam.mx/carlos/java/java_basico4_6.html)

y Sobrescritura de métodos en Java (Overriding)

<http://monillo007.blogspot.com/2008/01/sobrescritura-de-mtodos-en-java.html>,

enviar el código fuente de la clase que se especifica en el siguiente diagrama:



Agrega el método prender () y propón 2 sobre-cargas de éste.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



## Unidad III. Paradigma orientado a objetos



### ACTIVIDAD 2

Con la lectura de Sobreescritura de Métodos y TutorJava Nivel Básico, contesta las siguientes preguntas:

1. ¿Cuándo aplica la sobreescritura en Java?
2. ¿Qué reglas se debe seguir para sobreescibir?
3. ¿Las subclases son las únicas que pueden sobreescibir? ¿por qué?

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**

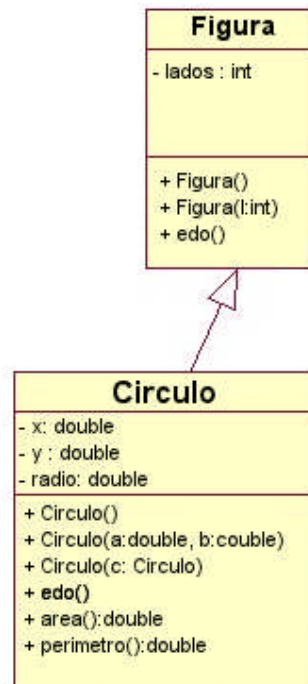


## Unidad III. Paradigma orientado a objetos



### Autoevaluación

Con base en el siguiente diagrama indique cuales de las afirmaciones son verdaderas y cuales son falsas. Al final obtendrás tu calificación de manera automática.



	Verdadera	Falsa
1 Figura hereda de Circulo	( )	( )
2 El método edo() se sobrescribe en Circulo	( )	( )
3 El método constructor de Circulo está sobrecargado	( )	( )
4 Si una clase hereda de Círculo no puede sobrescribir el método edo()	( )	( )
5 La sobrecarga de métodos aplica sobre el método edo()	( )	( )



## Unidad III. Paradigma orientado a objetos



### Tema 7. Encapsulamiento (ocultamiento de la información)

#### Objetivo del tema

Identificar los niveles de acceso a la información que pueden definir los objetos.

#### Desarrollo

La visibilidad se aplica a los atributos y a las operaciones de una clase, y a la misma clase también. Establece la proporción en que otras clases podrán utilizarlos. Existen tres niveles de visibilidad:

- Nivel público: la funcionalidad se extiende a otras clases. Para el nivel público se utiliza el lexema +
- Nivel privado: sólo la clase original puede utilizar el atributo u operación. Para el nivel privado se utiliza el lexema –
- Nivel protegido: la funcionalidad se otorga sólo a las clases que se heredan de la clase original. Para el nivel protegido se utiliza el lexema #

Lavadora
+marca: String -modelo: String #numeroSerie: Integer +capacidad: Integer
+agregarRopa() -sacarRopa() #agregarDetergente() +lavarRopa() +filtrarPelusa()

El encapsulamiento permite reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Así aumenta la cohesión de los componentes del sistema. Cada objeto está aislado del exterior y expone una interfaz a otros objetos que describe cómo pueden interactuar con él. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del





### Unidad III. Paradigma orientado a objetos



objeto pueden acceder a su estado. Esto asegura que otros objetos no puedan cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas.

#### ACTIVIDAD 1

En la página TutorJava Nivel básico

[http://www.programacion.com/java/tutorial/java\\_basico/17](http://www.programacion.com/java/tutorial/java_basico/17) se explican las palabras reservadas: public, private y protected. Elabora un cuadro sinóptico donde se explique las diferencias en el uso de estas palabras.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

#### Bibliografía básica

Autor	Capítulo	Páginas

#### Sitios electrónicos

Sitio	Descripción



## Unidad III. Paradigma orientado a objetos



### Tema 8. Diagrama de clases

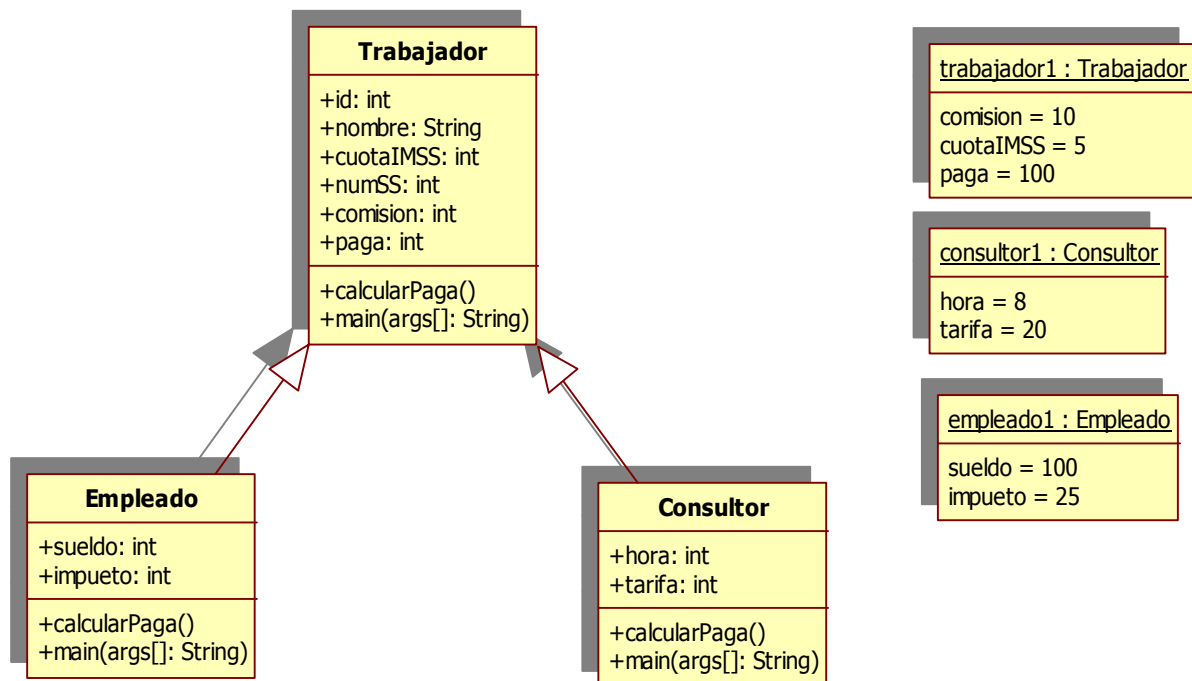
#### Objetivo del tema

Identificar los diagramas definidos en UML (Unified Modeling Language), para la representación de clases.

#### Desarrollo

Un diagrama de clases es una representación gráfica que modela a las clases y sus relaciones. La notación estándar para un diagrama de clases es la que define UML.

Ejemplo:





## Unidad III. Paradigma orientado a objetos



### ACTIVIDAD 1

Elabora el diagrama de clases correspondiente a las siguientes:

- Persona
- Alumno
- Trabajador
- Vehículo

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



### Unidad III. Paradigma orientado a objetos



#### Autoevaluación

Indica cuáles de las siguientes afirmaciones son verdaderas y cuales son falsas. Al final obtendrás tu calificación de manera automática.

	Verdadera	Falsa
1 En UML las clases especifican: nombre, atributos y métodos	( )	( )
2 En UML, el nivel de acceso a la información se especifica con los signos +, - y #.	( )	( )
3 Las relaciones entre clases se representan con notas.	( )	( )
4 La cardinalidad en las relaciones que se representan sólo sirven para indicar relaciones uno a muchos	( )	( )
5 Si una clase se conforma con otras, se debe indicar como una relación de agregación.	( )	( )

#### Bibliografía básica

Autor	Capítulo	Páginas

#### Sitios electrónicos

Sitio	Descripción





## Unidad III. Paradigma orientado a objetos



### Tema 9. Campos de aplicación

#### Objetivo del tema

Reconocer el uso actual del paradigma para el desarrollo de aplicaciones.

#### Desarrollo

Al igual que el paradigma imperativo, prácticamente con los lenguajes orientados a objetos se puede hacer cualquier cosa: controlar el plan de vuelo de una aeronave, aplicaciones para Web, aplicaciones gráficas, dispositivos móviles, videojuegos, realidad virtual, chips de memoria, etc. Algunos ejemplos de lenguajes son:

- Ada
- C++
- C#
- Delphi
- Eiffel
- Java
- Objective-C
- Oz
- PHP (en su versión 5)
- PowerBuilder
- Python
- Ruby
- Smalltalk
- VB.NET



## Unidad III. Paradigma orientado a objetos



### ACTIVIDAD 1

Genera un documento de no más de dos cuartillas donde indiques lo siguiente:

- Qué es UML.
- Reglas básicas para la generación de un diagrama de clases.
- Elabora el diagrama de clases para las clases Alumno, Maestro y Jefe de carrera.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

### ACTIVIDAD 2

Supón que vas a desarrollar un sistema para llevar la contabilidad de una farmacia. Desarrolla el diagrama de clases donde se muestren al menos seis clases (indica sus atributos y operaciones) y en dos de ellas aplica la relación de agregación.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



### ACTIVIDAD 3

Visita <http://www.youtube.com> y busca el video “*Java is everywhere*”  
<http://www.youtube.com/watch?v=guXCmQDy9Es> y  
<http://www.youtube.com/watch?v=Jg-vhAdtauE&feature=related>. Saca tus conclusiones de qué otras cosas se pueden hacer con Java además de videojuegos o aplicaciones para Web. Redacta el escrito en no más de una cuartilla.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

### ACTIVIDAD 4

Ejecuta el siguiente código y explica la diferencia entre Java y JavaScript. Los dos programas son OO, pero no se ejecutan de la misma forma:

```
<HTML>
<!--este es un ejemplo de POO-->
  <HEAD>
    <TITLE>EJEMPLO DEL HOLA MUNDO CON UN
    INTERPRETE Y UN COMPILADOR</TITLE>
  </HEAD>
  <BODY>
    <B>
      <CENTER>
```

Ejemplo de un programa interpretado y



### Unidad III. Paradigma orientado a objetos



compilado elaborado por:

```
<FONT COLOR='blue' SIZE="4">Ismael Perea</FONT>
</CENTER>
</B>
<SCRIPT LANGUAGE="JavaScript" type="text/javascript">
document.write("Hola Mundo esto es una prueba ");
</SCRIPT>
<P>
<APPLET width="208" height="50" CODE="HolaMundo.class"></APPLET>
<P>
Fin del ejemplo
</BODY>
</HTML>
```

*Este archivo se debe guardar con el nombre de holaMundo.html*

```
//
// Applet HolaMundo de ejemplo
//

/* éste es un ejemplo de un applet
en java para la clase de POO
elaborado por: Ismael Perea
*/

import java.awt.Graphics;
import java.applet.Applet;

public class HolaMundo extends Applet {
```





### Unidad III. Paradigma orientado a objetos



```
public void paint( Graphics g ) {  
    g.drawString( "¡Hola alumnos!",0,25 ) ;  
}  
}
```

*Este archivo se debe guardar con el nombre de `HolaMundo.java` y compilar con `javac`, y así obtendremos el archivo `HolaMundo.class`*

En un documento indica las diferencias de ejecución que detectaste al ejecutar estos programas.

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**



## Unidad III. Paradigma orientado a objetos



### Autoevaluación

Completa las siguientes frases arrastrando sobre la línea las palabras que completen correctamente cada oración. Al final obtendrás tu calificación de manera automática.

<input type="checkbox"/> Con la ___ se pueden abstraer los datos y la funcionalidad en objetos que al interactuar describen las acciones que se quieren programar.	<b>1</b> Programación orientada a objetos
<input type="checkbox"/> Los lenguajes orientados a objetos sintetizan las características y la funcionalidades de los objetos en las	<b>2</b> clases
<input type="checkbox"/> Con un diagrama de clases se pueden representar las ___ que existen entre las clases que se modelan.	<b>3</b> relaciones
<input type="checkbox"/> La programación orientada a objetos se pueden generar programas de índole _____	<b>4</b> general
<input type="checkbox"/> El lenguaje UML permite el análisis y _____ de las clases y objetos que intervendrán en un programa.	<b>5</b> diseño



## Unidad III. Paradigma orientado a objetos



### Cuestionario de Autoevaluación

Responde las siguientes preguntas.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

1. ¿Qué es el paradigma orientado a objetos?
2. ¿Qué es una clase?
3. ¿Qué es un objeto?
4. ¿Qué es un atributo?
5. ¿Qué es una operación?
6. ¿Qué es una relación?
7. ¿Qué es la responsabilidad de una clase?
8. ¿Qué es una dependencia?
9. ¿Qué es la generalización?
10. ¿Qué es el polimorfismo?



## Unidad III. Paradigma orientado a objetos



### Examen de Autoevaluación

Elige la opción que conteste correctamente cada pregunta. Al final obtendrás tu calificación de manera automática.

1. Esta relación es del tipo “tiene un”, o sea, un objeto del todo tiene objetos de la parte:

- a) Especialización
- b) Generalización
- c) Agregación
- d) Dependencia

2. Es una descripción de lo que hará la clase, es decir, de lo que sus atributos y operaciones intentan realizar en conjunto:

- a) Responsabilidad
- b) Herencia
- c) Polimorfismo
- d) Instancia

3. Una clase hija es una [...] de una clase padre:

- a) Generalización
- b) Agregación
- c) Composición
- d) Especialización



### Unidad III. Paradigma orientado a objetos



4. Es algo que la clase puede realizar, o que nosotros u otra clase pueden hacer a una clase. Su nombre se escribe en minúsculas si consta de una sola palabra; si son más, se unen iniciando con letra mayúscula:

- a) Atributo
- b) Operación
- c) Responsabilidad
- d) Nombre de la clase

5. Token que indica que un atributo u operación es privado:

- a) +
- b) -
- c) #
- d) \*

6. En la siguiente declaración: `import java.lang.System;` la clase es:

- a) Lang
- b) Java
- c) System
- d) Object

7. Token que indica que un atributo u operación es público:

- a) +
- b) -
- c) #
- d) \*



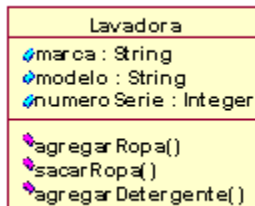
### Unidad III. Paradigma orientado a objetos



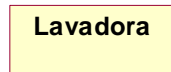
8. Es un tipo de relación cuando una clase utiliza a otra. Declara que un cambio en la especificación de un elemento puede afectar a otro elemento que la utiliza, pero no necesariamente a la inversa:

- a) Agregación
- b) Dependencia
- c) Composición
- d) Generalización

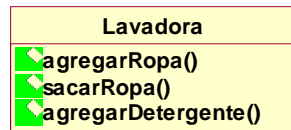
9. Representa una instancia de una clase:



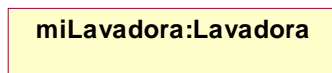
- a)



- b)



- c)



- d)



### Unidad III. Paradigma orientado a objetos



10. Propiedad o característica de una clase que describe un rango de valores que la propiedad podrá contener en las instancias:

- a) Operación
- b) Polimorfismo
- c) Herencia
- d) Atributo

#### Bibliografía básica

Autor	Capítulo	Páginas

#### Sitios electrónicos

Sitio	Descripción

#### LO QUE APRENDÍ

Identifica los conceptos principales vistos en esta unidad, después junto con tus compañeros formen equipos de 5 integrantes.

Cada equipo elaborará un mapa conceptual con los conceptos que identificaron cada uno de sus integrantes. Para identificar la participación de cada uno, utilicen un color o tipo de letra que los distinga.



## Unidad III. Paradigma orientado a objetos



### Glosario de la unidad

#### Objeto

Un objeto representa un ítem individual e identificable, o una entidad real o abstracta, con un papel definido en el dominio del problema. Un objeto tiene: Estado, comportamiento e Identidad.

#### Clase

Conjunto de objetos que comparten una estructura y comportamiento comunes. Una Clase representa una abstracción, la esencia que comparten los objetos. Las clases actúan como intermediarias entre una abstracción y los clientes que pretenden utilizar la abstracción





## Unidad III. Paradigma orientado a objetos



### **MESOGRAFÍA**

#### **Bibliografía básica**

#### **Bibliografía complementaria**

#### **Sitios electrónicos**



### Unidad III. Paradigma orientado a objetos



(ANEXO 1) [Descargar archivo Horas PDF](#) que se encuentra en la plataforma