



## **Manual de prácticas de laboratorio**

Título: **Análisis de Sistemas**

Para la Unidad de Aprendizaje: **Análisis de Sistemas**

Créditos de la Unidad de Aprendizaje: **8**

Espacio académico en que se imparte la UA: CU UAEM Valle de Chalco

Versión 1. Noviembre 2013

### **Datos de Identificación**

Programa Educativo: Ingeniería en Computación

Programa de Estudios por Competencias :Análisis de Sistemas

Unidad de Competencia 3:

Elaborado por:

**Autor:**

**M. en T.E. Marisol Hernández Hernández**

**coautor**

**M. en I. Marco Alberto Mendoza Pérez**

**M. en T.I Rodolfo Melgarejo Salgado**

Agosto del 2018



## ÍNDICE

Datos de Identificación	1
Presentación	3
Estructura de la Unidad de Aprendizaje	4
Propósito de la Unidad de Aprendizaje	4
Prácticas:	
1. Diagrama de objetos	7
2. Diagrama de clases	13
3. Polimorfismo	20
4. Herencia	25
5. Referencias	30





## **Presentación**

Una de las tareas a las que se enfrenta todo Ingeniero en Computación en su desempeño profesional es al desarrollo de aplicaciones automatizadas, que resuelvan problemas específicos de manejo de información en las organizaciones.

La complejidad de los problemas a resolver, la dificultad misma para definirlos adecuadamente y las presiones de tiempo y costos que se ejercen sobre los proyectos informáticos, requieren de la aplicación de metodologías de desarrollo de aplicaciones que puedan garantizar, dentro de ciertos parámetros, software de calidad, fácilmente adaptable y mantenible, que apoye realmente la toma de decisiones de las organizaciones para el logro de sus objetivos.

Las metodologías para desarrollo de sistemas involucran necesariamente una etapa de planeación del proyecto y análisis de la organización, y del entorno donde habrá de instrumentarse la solución de manejo de información.

El análisis es una etapa fundamental para el desarrollo de aplicaciones que involucra la generación de modelos bajo diferentes paradigmas, que pueden aplicarse dependiendo del contexto del problema a solucionar. Es importante que el ingeniero en computación evalúe y aplique diferentes enfoques teórico/prácticos y sus herramientas asociadas de acuerdo a las tendencias actuales de desarrollo de sistemas.

Un buen modelo de análisis es la base que puede garantizar el diseño y construcción de software de calidad. El análisis de sistemas constituye la parte medular de la ingeniería de software, siendo parte fundamental en la formación práctica del ingeniero en computación.

El análisis orientado a objetos involucra elementos con los que se pueden representar los requerimientos y funcionamiento de un sistema de información, elementos como las clases que representan las entidades de los sistemas y que a su vez pueden derivar a elementos más específicos como los objetos, son ideales para el análisis de sistemas. Además características propias como son el polimorfismo que significa el comportamiento de múltiples formas de cada método que se deriva de las clases, o la herencia, con la cual se puede hacer uso del reciclaje de elementos a los cuales puede agregarse atributos y comportamiento a los ya heredados de su clase base, hace de este manual ideal para aprender una parte esencial del análisis de sistemas.





Cuando el alumno desarrolla estas prácticas comprende mejor la abstracción de los conocimientos que se necesitan saber para el desarrollo de la metodología orientada a objetos. Con el diagrama de objetos se enseña a representar los elementos específicos de las clases así como sus interacciones. Cuando hace la práctica de las clases aprende a identificar los elementos que las componen (métodos y atributos), así como sus relaciones, y por último, con los temas de polimorfismo y herencia de clases, que son temas de especialización del diseño orientado a objetos tan abstractos que si se representan en diagramas de UML como se hace en las prácticas, los alumnos pueden comprender mejor su definición y aplicación.

### **Estructura de la Unidad de Aprendizaje**

1. Adquirir la capacidad para planear el desarrollo de proyectos de ingeniería de software
2. Conocer y aplicar el paradigma estructurado para construir modelos de análisis de sistemas de información en las organizaciones
3. Conocer y aplicar el paradigma orientado a objetos para construir modelos de análisis de sistemas de información en las organizaciones
4. Aplicar herramientas automatizadas que faciliten la planeación de proyectos de ingeniería de software y la construcción de modelos





PROGRAMA DE ESTUDIO POR COMPETENCIAS  
ANÁLISIS DE SISTEMAS

I. IDENTIFICACIÓN DEL CURSO

ESPACIO EDUCATIVO: Facultad de Ingeniería						
LICENCIATURA: Ingeniería en Computación				ÁREA DE DOCENCIA: Programación e Ingeniería de Software		
AÑO DE APROBACIÓN POR EL CONSEJO UNIVERSITARIO:						
APROBACIÓN POR LOS HH. CONSEJOS ACADÉMICO Y DE GOBIERNO		FECHA:		PROGRAMA ELABORADO POR: Dr. José Antonio Álvarez Lobato Mtra. Tania Lilia Chávez Soto		PROGRAMA REVISADO POR: Integrantes de la Academia de Programación e Ingeniería de Software  Centro Universitario Valle de Chalco
				FECHA DE ELABORACIÓN : 11 de Noviembre 2013		FECHA DE REVISIÓN : Noviembre 2013
CLAVE	HORAS DE TEORÍA	HORAS DE PRÁCTICA	TOTAL DE HORAS	CRÉDITOS	TIPO DE CURSO	NÚCLEO DE FORMACIÓN
L41008	3 horas	2	5	8	Curso	Sustantivo
UNIDAD DE APRENDIZAJE ANTECEDENTE Teoría de Sistemas				UNIDAD DE APRENDIZAJE CONSECUENTE Diseño de Sistemas		
PROGRAMAS EDUCATIVOS O ESPACIOS ACADÉMICOS EN LOS QUE SE IMPARTE: Licenciatura en Ingeniería en Computación (Facultad. de Ingeniería, Centros Universitarios: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco, Valle de México, Valle de Teotihuacán, Zumpango)						

**Propósito**

Conocer y aplicar distintas metodologías aplicadas en la planeación y análisis de sistemas.





### Unidad de competencia III.

#### Paradigma orientado a objetos

Conocer el paradigma orientado a objetos como marco teórico conceptual para construir modelos de análisis de sistemas de información en las organizaciones.

UNIDAD DE COMPETENCIA III	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Conocer el paradigma orientado a objetos como marco teórico conceptual para construir modelos de análisis de sistemas de información en las organizaciones.	Tecnología de objetos	Capacidad de abstracción	Responsabilidad en el uso y destino de información interna de las organizaciones
	Principios de la orientación a objetos (abstracción, jerarquía, encapsulamiento) <b>Objeto, clase, polimorfismo, herencia.</b> Metodologías de análisis orientado a objetos	Capacidad para recopilar información de manera sistemática  Capacidad de síntesis e interpretación de la información  Capacidad de comunicación	Tolerancia a la crítica Responsabilidad en la entrega oportuna de materiales escritos
		oral y escrita  Capacidad de investigación	
<b>ESTRATEGIAS DIDÁCTICAS:</b> • Recomendación y discusión de lecturas • Investigación en Internet • Lluvia de ideas		<b>RECURSOS REQUERIDOS</b> Pizarrón Video proyector Computadora	<b>TIEMPO DESTINADO</b> 10 hrs.
CRITERIOS DE DESEMPEÑO III	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	
Investigación de los avances recientes en la aplicación de la tecnología de objetos para el desarrollo de aplicaciones	Concepto, tecnología o herramienta investigada  Fuentes consultadas	Reportes de Investigación documental	
Identificación adecuada de los conceptos de la Tecnología de Objetos y de los procesos de análisis orientado a objetos	Respuestas adecuadas a los cuestionamientos sobre los conceptos y principios de la orientación a objetos  Fuentes consultadas  Respuestas a cuestionamientos	Evaluación escrita	





## Práctica 1: Diagrama de Objetos

### Introducción

Los objetos son personas, lugares o cosas que son relevantes para el sistema bajo análisis.

Los

objetos podrían ser clientes, artículos, pedidos, etc. Los objetos también podrían ser pantallas

GUI o áreas de texto en la pantalla (Kendal y Kendal, 2011).

Son Instancia de una clase, es decir entidades provistas de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), que reaccionan a eventos. En el mundo real hay infinidad de objetos, ejemplo: de la clase animal, se puede tener un objeto: perro o gato De la clase gente, se puede tener un objeto: Luis o Tobías.

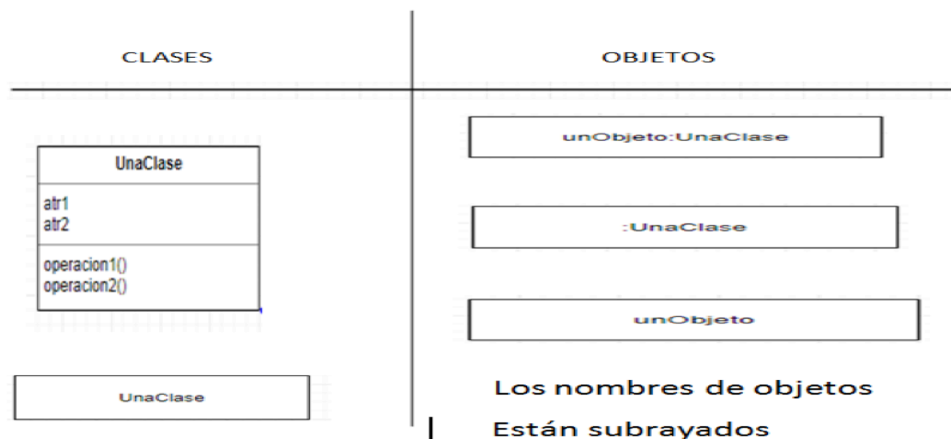
Los diagramas de objetos se utilizan para visualizar las estructuras de datos y las interacciones que existen entre objetos en tiempo de ejecución y se derivan de la definición de las clases.

Los objetos:

- Muestra instancias compatibles con un diagrama un diagrama de clase particular. de clase particular.

- Incluye sus objetos y los valores de sus datos. Incluye sus objetos y los valores de sus datos.

En los diagramas UML un objeto se representa en cuadros, el nombre se escribe con mayúsculas empezando con dos puntos y las relaciones se representan con líneas sin dirección.





## Descripción

Representa mediante un diagrama de objetos la siguiente especificación:

Crear una clase Persona con los atributos nombre y edad de tipo de visibilidad privado y un método mostrar de tipo público. Esta clase heredará a las clases Empleado con atributo privado llamado sueldo bruto y 2 métodos públicos que son mostrar y calcular salario neto y la segunda clase heredada se llama cliente con el atributo privado teléfono de contacto y el método público, mostrar, agregar una clase que herede de empleados (Directivo) con el atributo privado llamado categoría y de la clase cliente heredar a Empresa con el atributo privado nombre.

Agregar su diagrama de objetos

**Duración:** 2 horas

**Objetivos de aprendizaje:** EL alumno aprenderá y aplicará la forma de crear diagramas de Objetos

## Requisitos Previos

Tener conocimientos sobre lenguaje UML.

## Materiales

Computadora

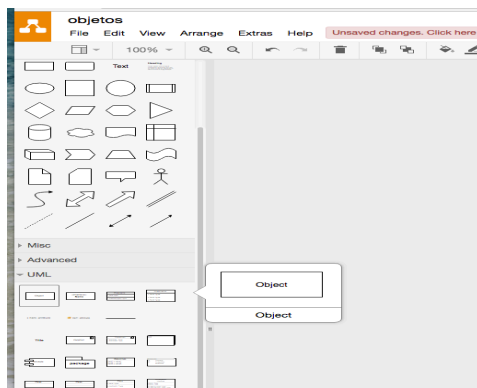
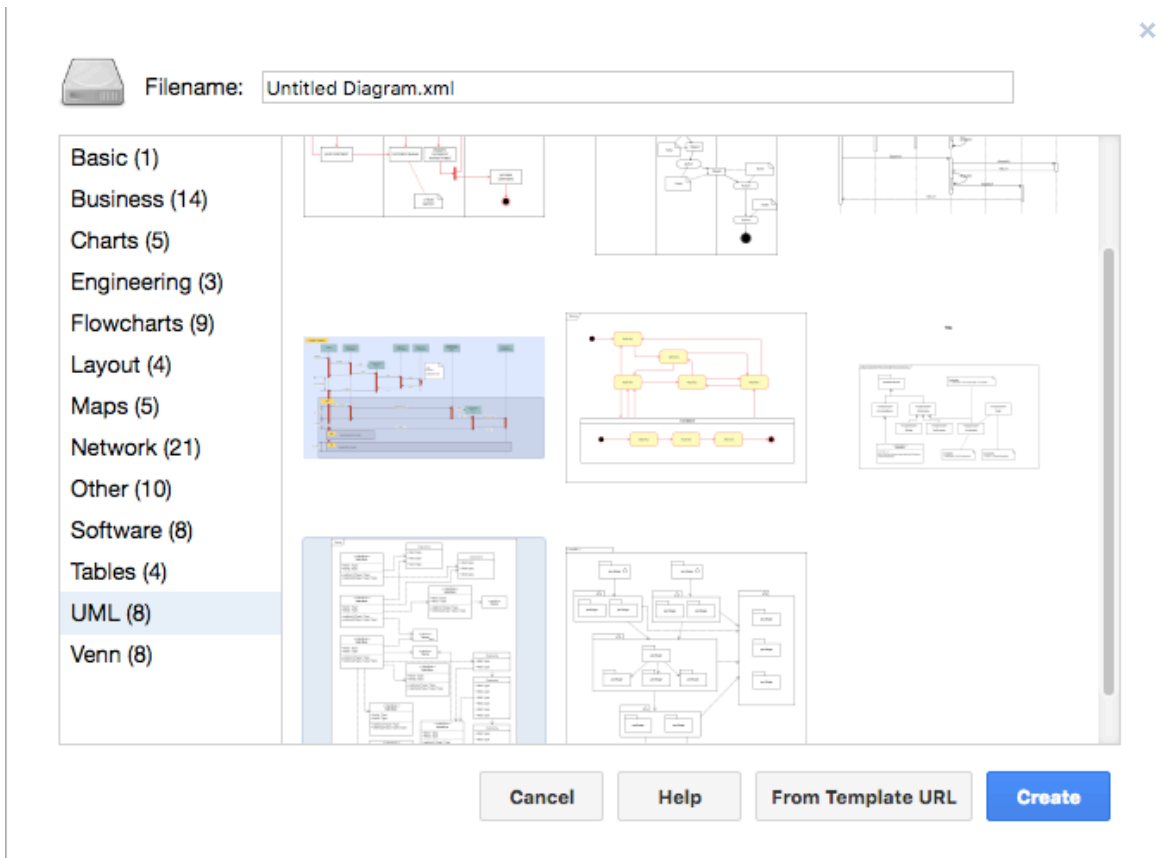
internet

## Procedimiento

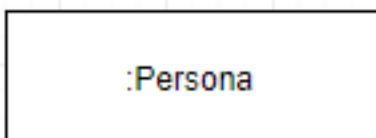
- 1.- Ingresar al sitio <https://www.draw.io/>
2. Crear un nuevo diagrama
3. Seleccionar el diagrama de objetos
4. Modificar y en su caso agregar o eliminar objetos







1. A partir de una clase **Persona** con los atributos nombre y edad de tipo de visibilidad privado y un método mostrar de tipo público. **Crear su objeto (no olvidar empezar el nombre con : y letra mayúscula según la nomenclatura de UML)**





### Figura 1. Creación de un objeto

2. La clase heredará a las clases Empleado y cliente con atributos privado llamado sueldo bruto y 2 métodos públicos que son mostrar y calcular salario neto, **crear sus objeto.**

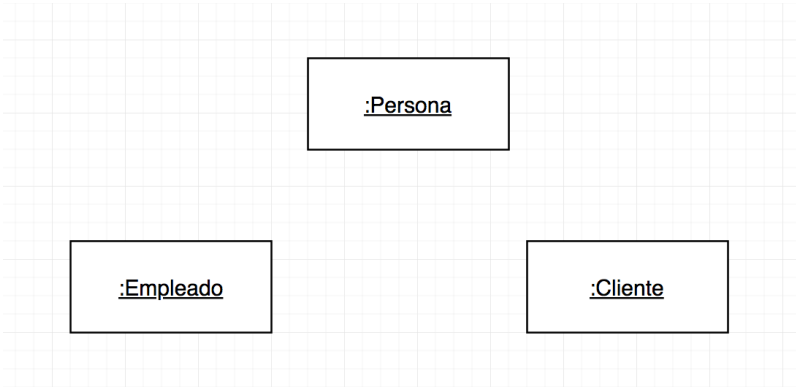


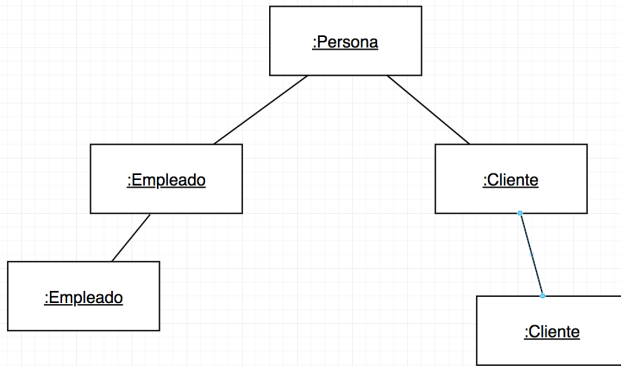
Figura 1. Creación de 2 objetos

1. La clase Empleado heredada a la clase Directivo y cliente hereda a la clase empresa. Crear sus objetos

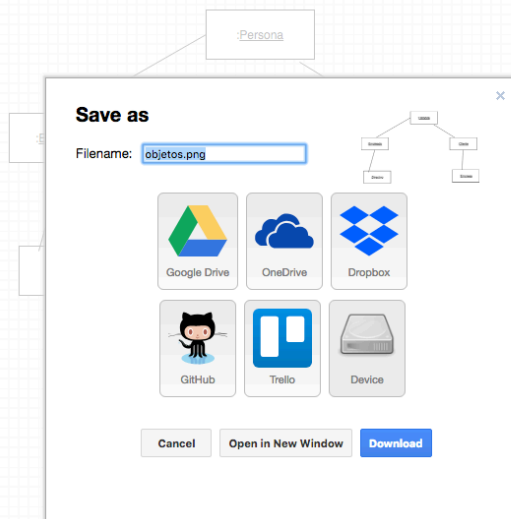


2. Conectarlos según lo especifica el problema.





### 3. Exportar la imagen



### Síntesis

El análisis y diseño orientado a objetos modela sistemas que interactúan entre sí con elementos funcionales compuestos por verbos y sustantivos, como por ejemplo: El empleado tiene (atributos), El empleado hace(métodos), estos elementos son llamados objetos y son esenciales en el análisis, además un sistema puede ser escalable utilizando objetos.

Los diagramas orientados a objetos sirven para representar la complejidad de los requerimientos de los sistemas, las entidades que representan son especificaciones de las clases. Los objetos contienen los métodos y atributos de una clase y su relación depende de la herencia de clases.



### **Preguntas de Repaso**

1. De la siguiente oración, señale cual es el objeto:  
Edgar es un alumno
2. Cual es el tipo de letra con la que debe empezar el nombre de un objeto
3. De qué depende que un objeto tenga ciertos atributos y métodos?
4. Dibuje un diagrama de objetos de la jerarquía animales que hereda a los animales mamíferos y obtenga 3 los objetos de la clase animales mamíferos.
5. Los diagramas de objetos representan una perspectiva del sistema, es cierto o falso?



## Práctica 2: Diagrama de clases

### Introducción

En ingeniería de software un diagrama de clases es una estructura que muestra las clases del sistema con sus atributos y métodos, en donde pueden tener una visibilidad protegida, privada o pública y que sirven para dar una perspectiva de la funcionalidad del sistema en sus relaciones con los objetos que se generan a partir de esas clases. Las clases permiten al programador modelar objetos que tienen atributos (representados como datos miembro) y comportamientos u operaciones (representadas como funciones miembro) (Deitel y Deitel, 2008). Los elementos de una clase se representan según el diagrama de la siguiente figura:

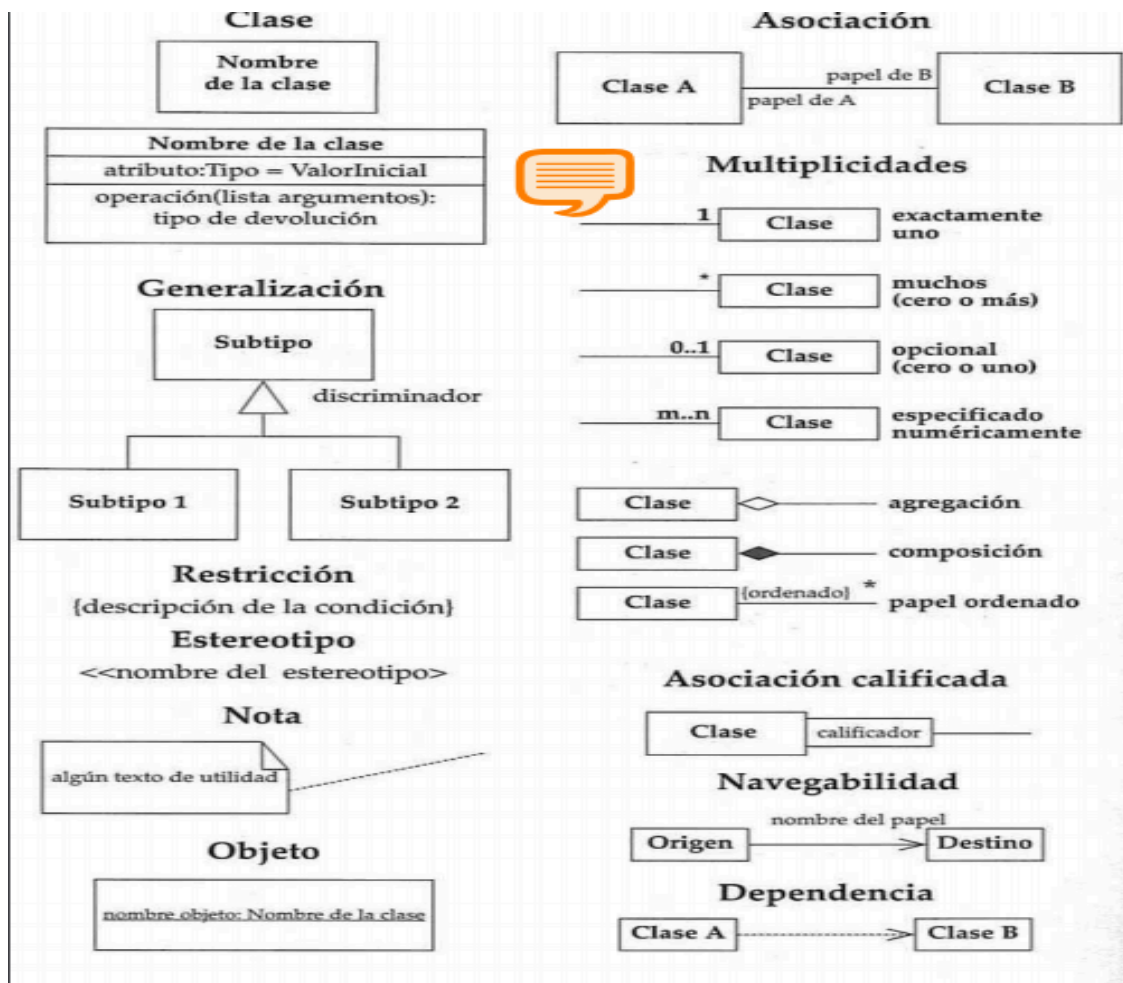


Diagrama de clases (Fowler y Scott, 1997)

En donde se puede observar que las clases reúnen los atributos y métodos de un objeto, por ejemplo la clase animales contendrá los atributos de raza, tipo, color y los métodos de comer, dormir, correr etc.



Los atributos y los métodos de las clases pueden ser:

+ Público

- Privado

# Protegido

/ Derivado

~ Paquete

Las clases se relacionan con asociaciones con el siguiente significado:

Las asociaciones de agregación se representa con rombo hueco y representa una relación de tipo parte-todo o parte-de.

Las asociaciones de composición se representa con diamante color negro y significa que los componentes constituyen una parte del objeto compuesto.

Las asociaciones de generalización se representa una flecha hueca y significa herencia

Los diagramas de clases tienen multiplicidad de 0 a 1, de 0 a muchos o de n a m (números exactos).

También puede agregarse comentarios sobre la clase y se representa con un cuadro aparte que se les llama nota

### **Descripción de la práctica**

Crear una clase Persona con los atributos nombre y edad de tipo de visibilidad privado y un método mostrar de tipo público. Esta clase heredará a las clases Empleado con atributo privado llamado sueldo bruto y 2 métodos públicos que son mostrar y calcular salario neto y la segunda clase heredada se llama cliente con el atributo privado teléfono de contacto y el método público mostrar, agregar una clase que herede de empleados (Directivo) con el atributo privado llamado categoría y de la clase cliente heredar a Empresa con el atributo privado nombre.

Agregar una multiplicidad de Directivo a empleados y una agregación en donde el cliente tenga una empresa y una composición en donde a empresa tenga empleados.

**Duración:** 2 horas

**Objetivos de aprendizaje:** El alumno aprenderá y aplicará la forma de crear gráficos de clases.





## Requisitos Previos

Tener conocimientos sobre lenguaje UML.

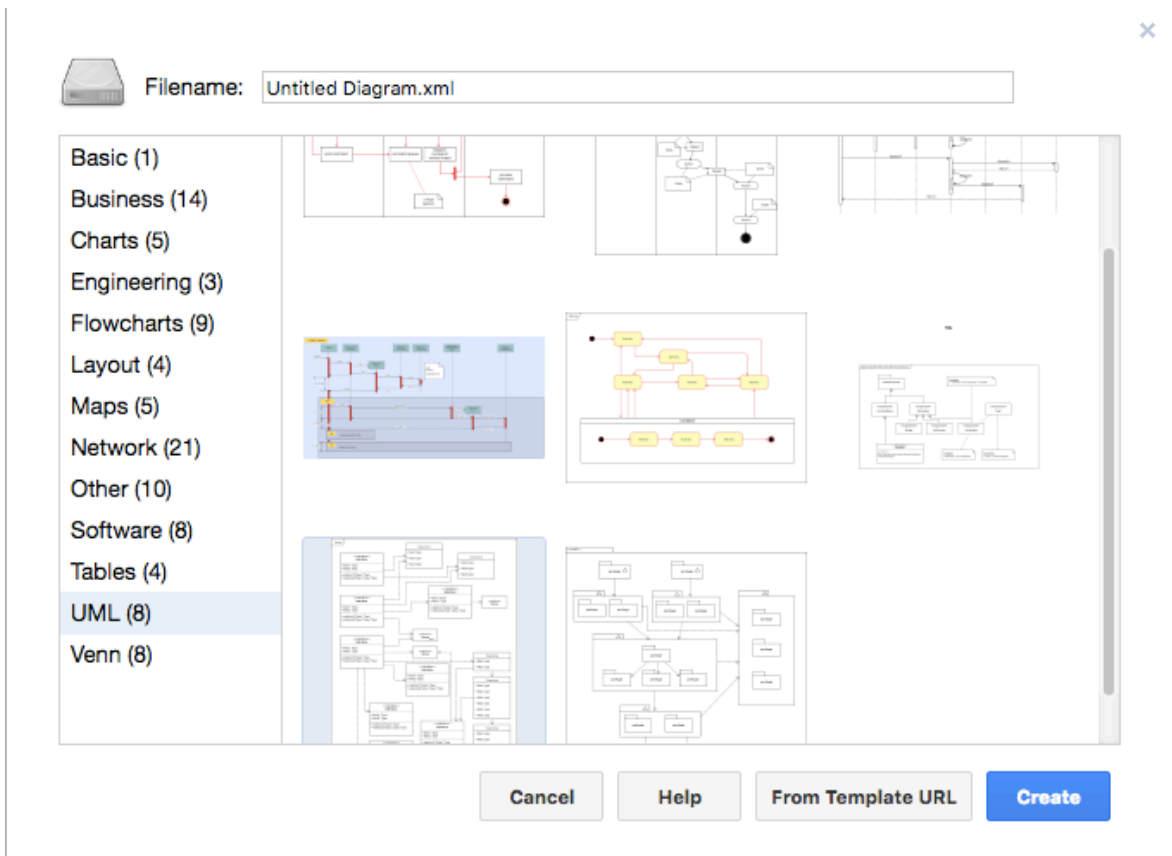
## Materiales

Computadora

internet

## Procedimiento

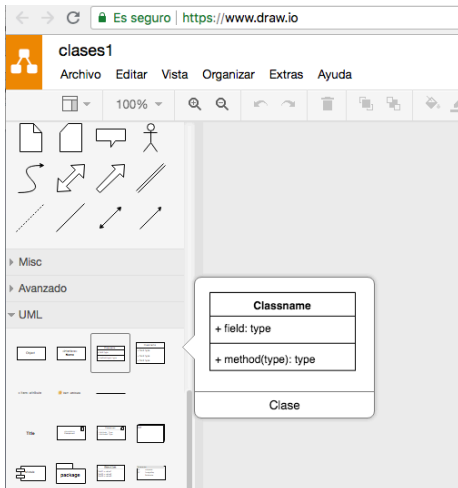
- 1.- ingresar al sitio <https://www.draw.io/>
- 2.- Seleccionar el tipo de diagrama clase



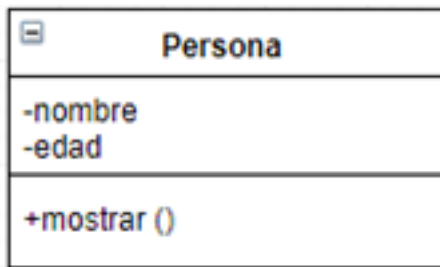
## Procedimiento

3. Modificar el diagrama o en su caso agregar más clases del menú UML que se encuentra del lado izquierda.

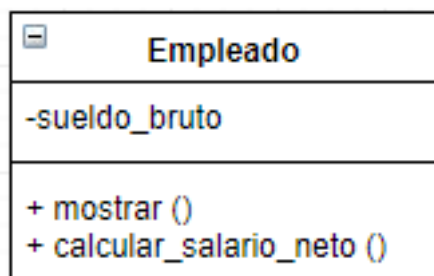




4. Agregar la clase Persona con los atributos nombre y edad de tipo de visibilidad privado y un método mostrar de tipo público.

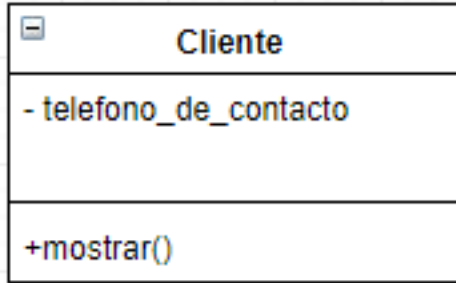


6. Esta clase heredará a las clases Empleado con atributo privado llamado sueldo bruto y 2 métodos públicos que son mostrar y calcular salario neto.

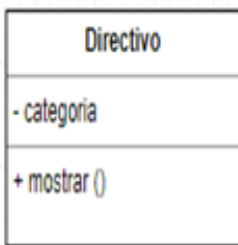


5. La segunda clase heredada se llama cliente con el atributo privado teléfono de contacto y el método público mostrar,

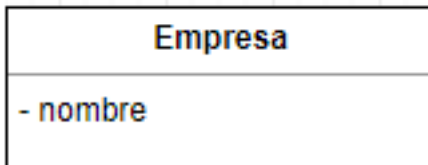




6. agregar una clase que herede de empleados (Directivo) con el atributo privado llamado categoría.

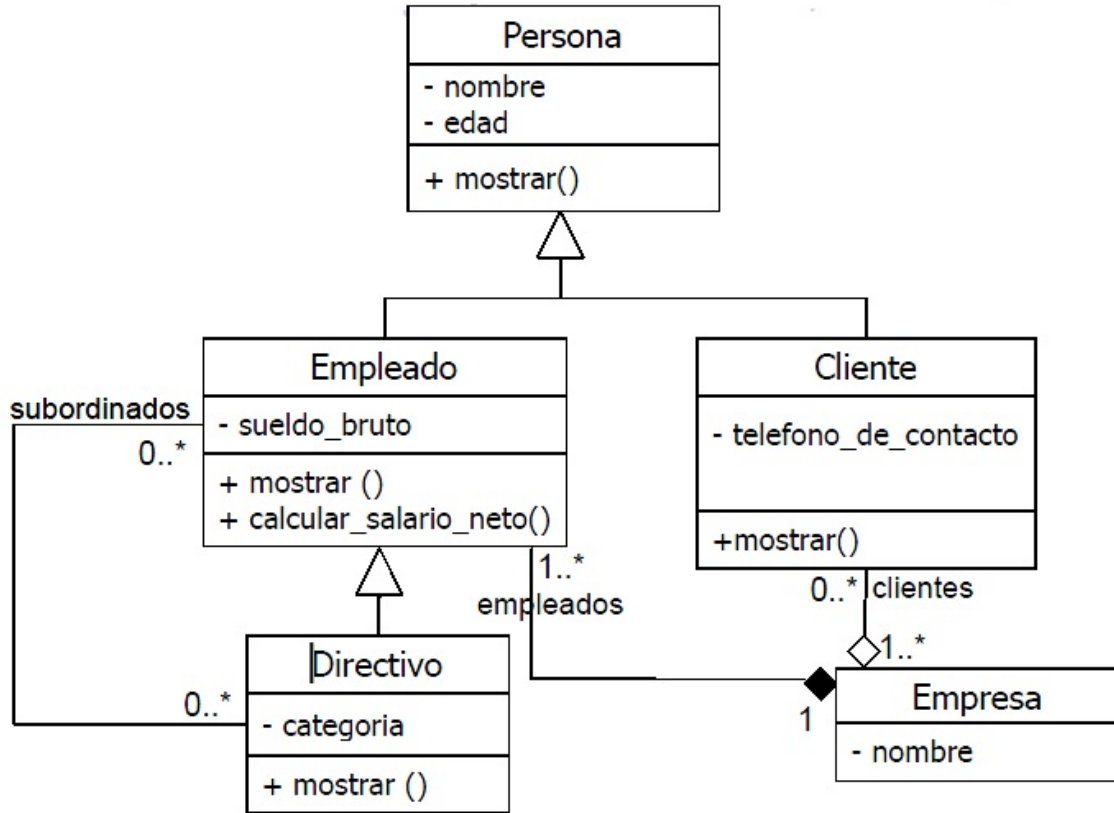


7. De la clase cliente heredar a Empresa con el atributo privado nombre.

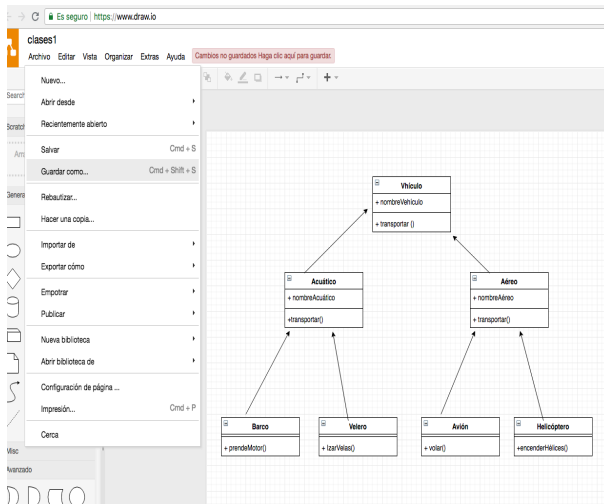


8. Agregar una multiplicidad de Directivo a empleados y una agregación en donde el cliente tenga una empresa y una composición en donde a empresa tenga empleados.





6.- Guardar y exportar en formato de imagen y agregarlo en la ubicación que desees archivarlo.



7. Descargar la imagen





## Síntesis

Los diagramas de clases son una manera de representar una perspectiva general de los sistemas de información, con ellos mostramos la manera en que actúan las entidades de los sistemas, incluyendo las relaciones interactúan. Los diagramas de clases son una manera de representar el análisis del sistema, en ellos se puede representar las asociaciones para mayor comprensión por parte del alumno de

## Preguntas de Repaso

1. Anota el significado y símbolo de la asociación llamada composición
2. Anota el significado y símbolo de la asociación llamada Agregación
3. Anota el significado y símbolo de la asociación llamada Generalización
4. Representa una asociación de 0 a muchos
5. Representa una asociación de n a m





## Práctica 3 Polimorfismo

### Introducción

EL polimorfismo que significa muchas formas o redefinición de método, es la capacidad de un programa orientado a objetos para tener varias versiones del mismo método con el mismo nombre dentro de una relación superclase/subclase. Por ejemplo la subclase podría heredar un método de su clase madre, pero podría agregar o modificar el comportamiento. (Kendal y Kendal, 2011).

El polimorfismo es la propiedad que permite enviar mensajes sintácticamente iguales a diferentes tipos de objetos, en dicho proceso la condición es saber responder al mensaje que se envía de manera polimórfica.

La manera en que se envía depende el tipo de lenguaje que se esté utilizando, por ejemplo puede un objeto puede comportarse de una manera enviando el mensaje con el método abrir ventanas o abrir archivos.

### Descripción de la práctica

Suponga que tiene 3 clases Figura, y triángulo, y cuadrado, todos tienen un método que los dibuja. Represente el polimorfismo en un diagrama de clases

**Duración:** 2 horas

**Objetivos de aprendizaje:** El alumno aprenderá a analizar y plasmar los conocimientos abstractos en un diagrama de objetos utilizando polimorfismo.

### Requisitos Previos

Saber la teoría de el desarrollo de un diagrama de clases  
Saber términos sobre programación orientada a objetos  
Tener conocimientos de UML.

### Materiales

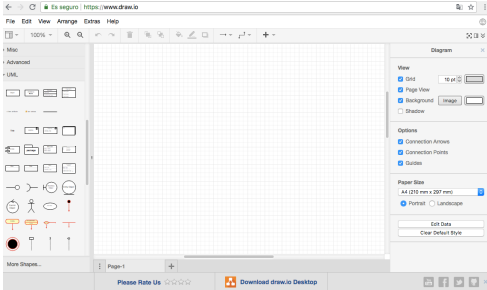
Computadora  
internet  
accesos a Draw.io

### Procedimiento

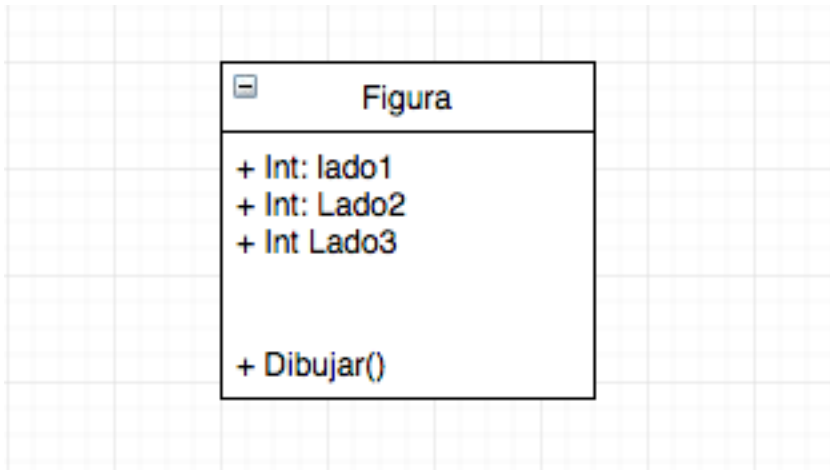


1.- Ingresar al sitio <https://www.draw.io/>

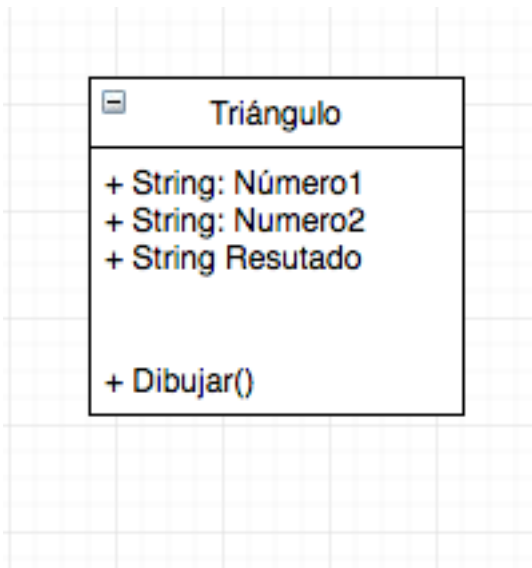
2.- Seleccionar el tipo de diagrama clase, o diagrama en blanco



3. Agregar las clases figura

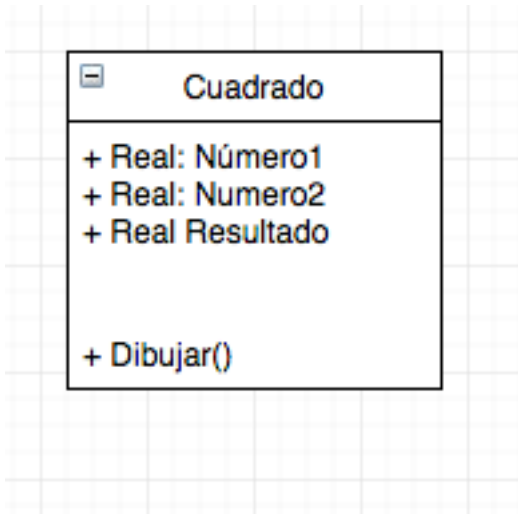


3. Agregar las clases Triángulo



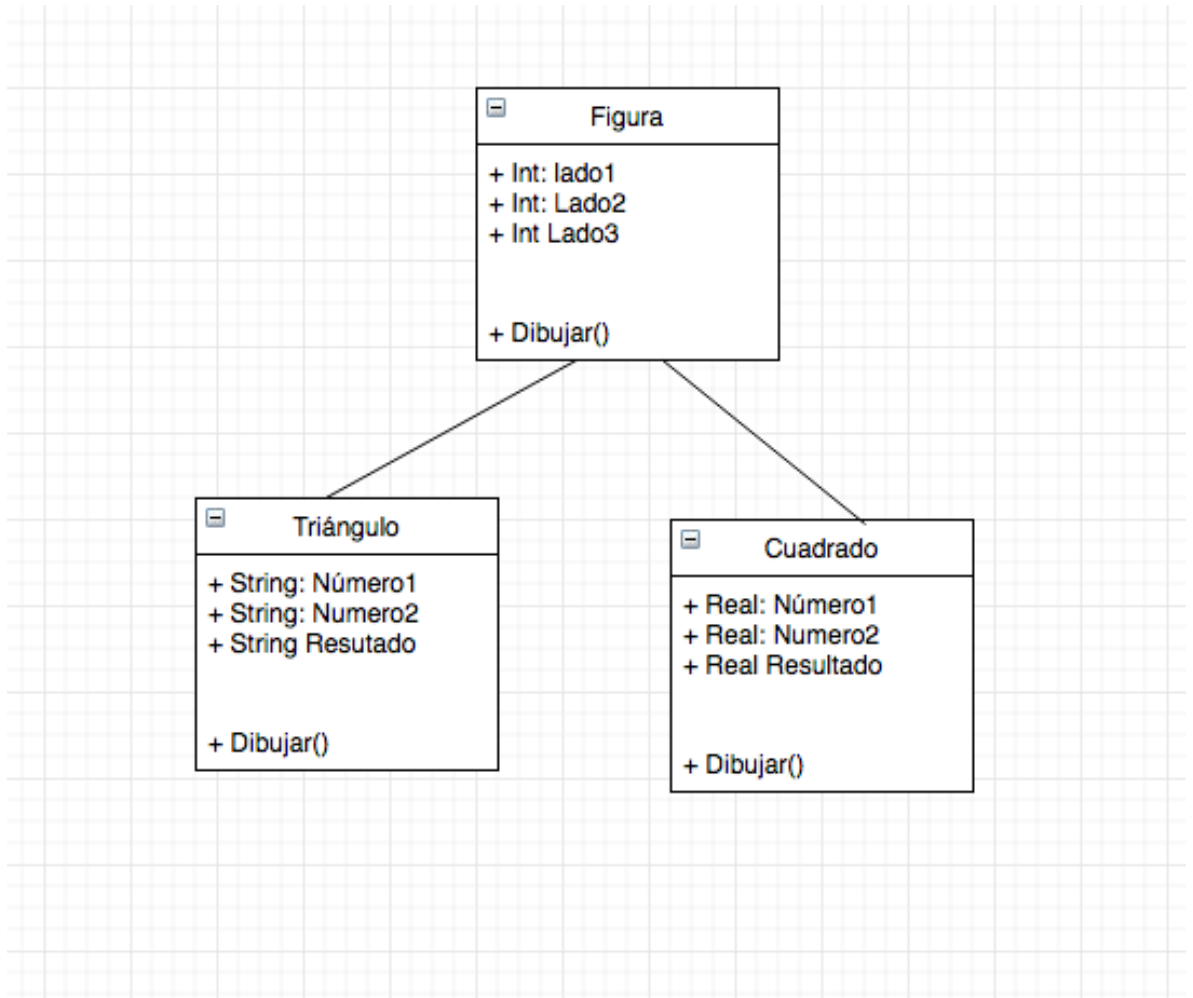


3. Agregar las clases Cuadrado

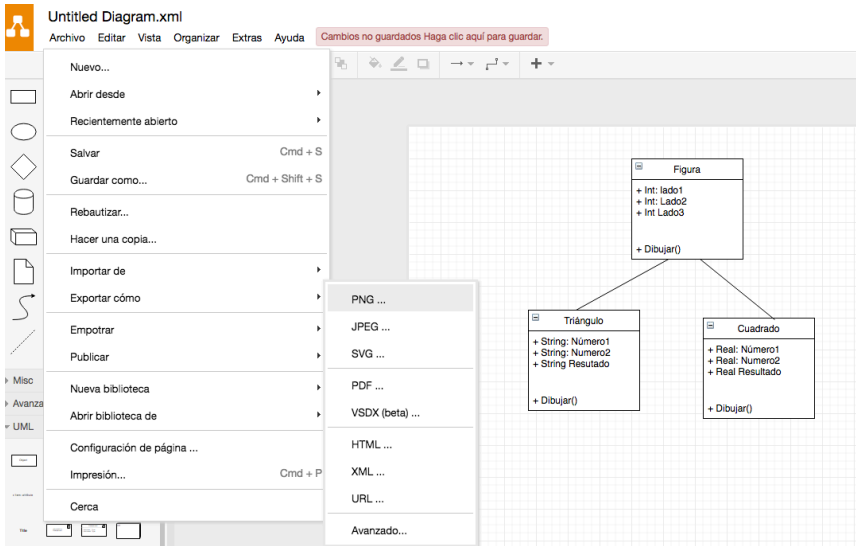


6. Agregarlas para que hereden de la clase Figura





### 7. Descargar la imagen como tipo jpg o png





### **Síntesis**

Los diagramas de clases que representan el polimorfismo sirven para representar las diferentes formas en que puede funcionar un objeto, por ejemplo: el término abrir, puede representar una ventana, un archivo, una base de datos, ese término aunque es el mismo adquiere diferentes formas dependiendo el objeto por el que es llamado

### **Preguntas de Repaso**

1. Que datos necesitará sean enviados al método dibuja en el caso del triángulo
2. Dibuja una clase con su método dibujar que se llame circunferencia y agrega los datos necesarios para dibujarla.
3. Si se deseara agregar comportamiento a la clase triángulo, que le agregarías?, dibújala.
4. Dibuja el polimorfismo para la clase animales y sus clases mamíferos y herbívoros.





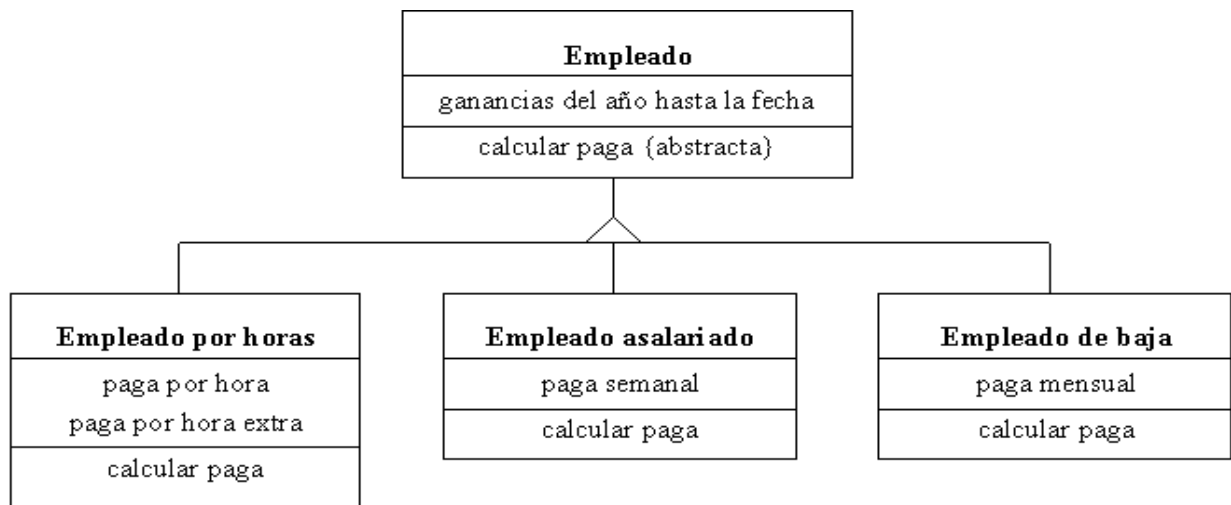


## Práctica 4: Herencia

### Introducción

Es el mecanismo jerárquico que facilita un proceso de generalización-especialización de las clases involucradas en un sistema. Por ejemplo, un automóvil es la especialización de una clase llamada vehículo y consecuentemente, el automóvil conserva las características generales de ser un vehículo (Cortés, 1998).

La herencia permite a una clase heredar los atributos y métodos de otra clase, facilitando de esta forma la reutilización del código y permitiendo crear nuevas clases mediante la abstracción de los atributos y métodos comunes.



Las clases Empleado por horas, empleado asalariado y empleado de baja, heredan de la clase Empleado, esto significa que la clase Empleado por horas tiene los siguientes atributos:

Paga por hora, paga por hora extra y calcula paga, pero además tiene los atributos que hereda como ganancias del año hasta la fecha y calcular paga (abstracta)

### Descripción de la práctica

Representar el diagrama que soluciona el siguiente problema:

Representar una Clase Padre con el nombre: Vehículo, la cual tiene como hijas a las clases Acuático y Aéreo, estas a su vez son





clases Padre de Barco, Velero, Avión y Helicóptero que a la vez derivadas de las derivadas de Vehículo.

La clase Avión puede usar el método volar() de la clase Aéreo, así como también usar el método transportar() de la clase Vehículo, pues se encuentra en el mismo árbol de herencia, sin embargo no puede usar el método navegar() de la clase Acuático, ya que no es de tipo Acuático.

Se enuncia un problema y a partir de ello, el alumno deberá elaborar el diagrama de clases con algún software de ingeniería de software.

En este caso se realizará con Draw.io, que es un software libre en línea, en el que se pueden diseñar muchos tipos de diagramas, y entre ellos los diagramas UML  
<https://www.draw.io/>

**Duración:** 2 horas

**Objetivos de aprendizaje:** El alumno aprenderá a analizar y plasmar los conocimientos abstractos en un diagrama de clases

### **Requisitos Previos**

Saber la teoría de el desarrollo de un diagrama de clases  
Saber términos sobre programación orientada a objetos  
Tener conocimientos de UML.

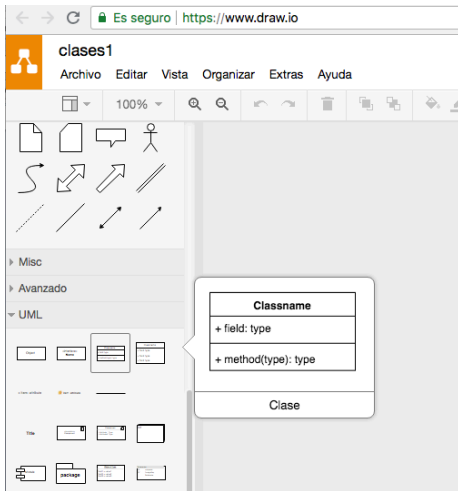
### **Materiales**

Computadora  
internet  
accesos a Draw.io

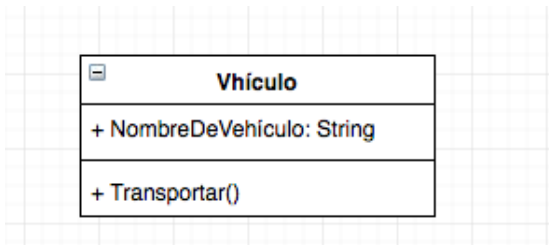
### **Procedimiento**

- 1.- Ingresar al sitio <https://www.draw.io/>
- 2.- Seleccionar el tipo de diagrama clase, o diagrama en blanco
3. Agregar las clases del menú UML que se encuentra del lado izquierda

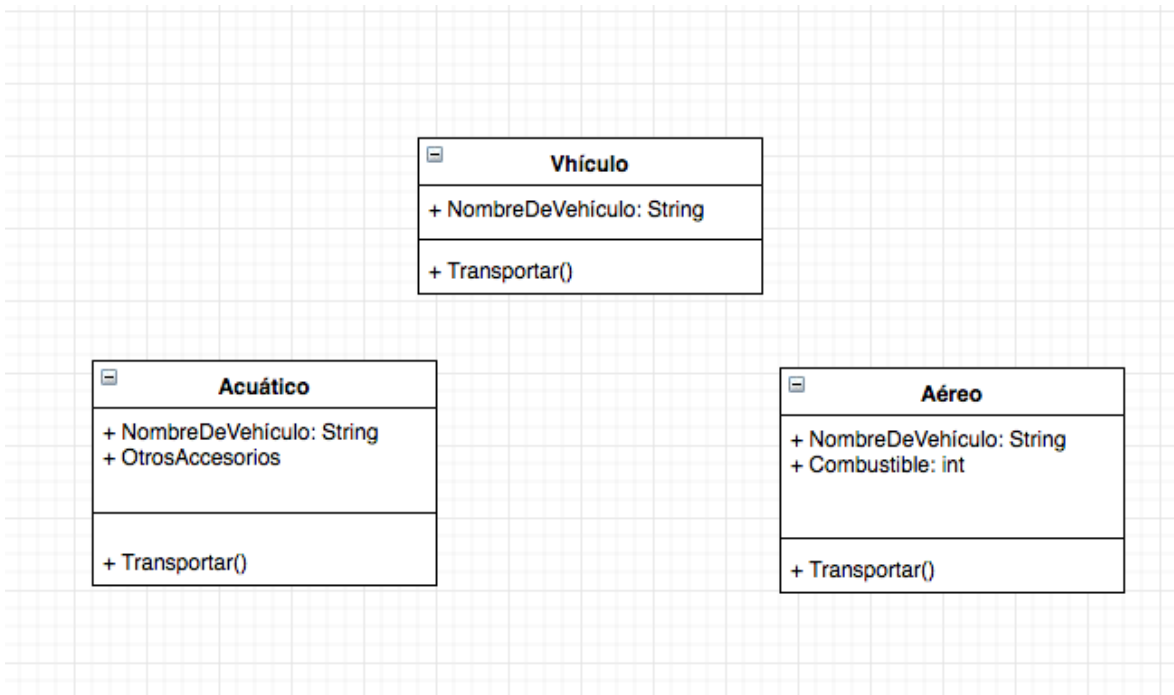




4. Representar una Clase Padre con el nombre: Vehículo.

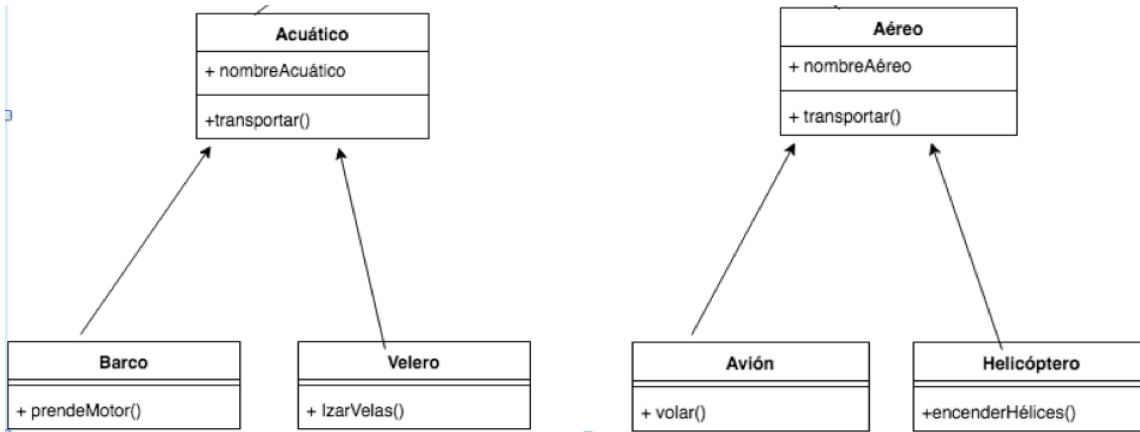


5. Agregar las clases Acuático y Aéreo

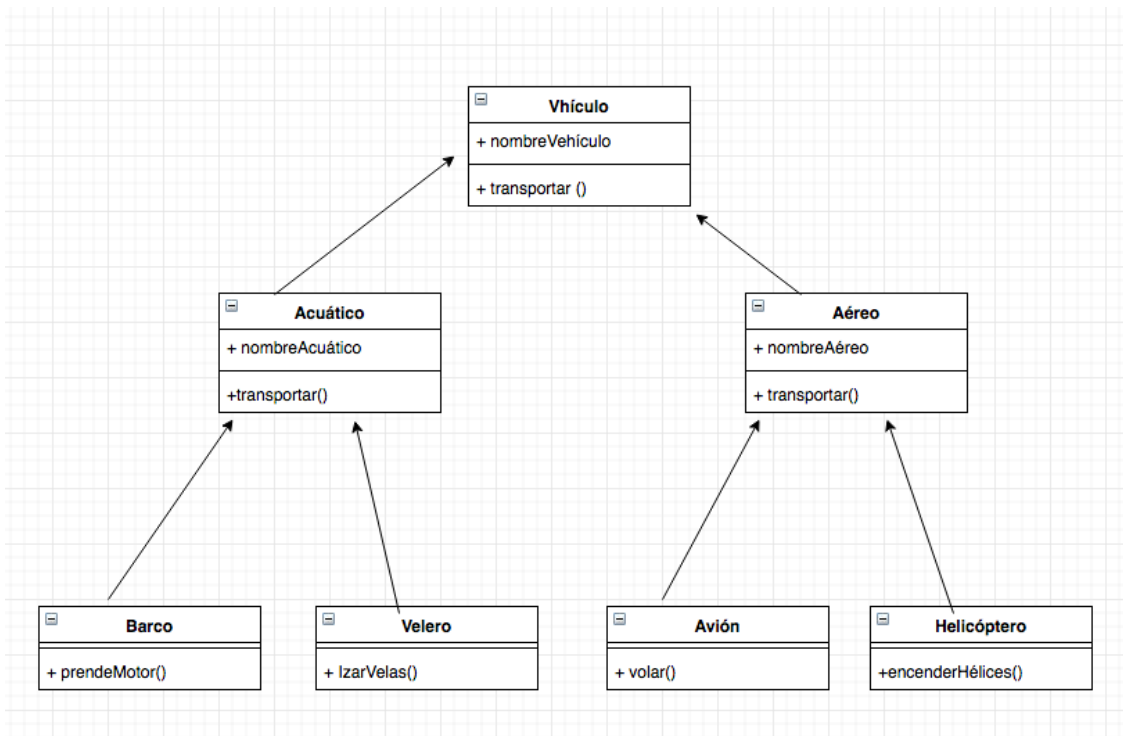


8. Agregar las clases y sus respectivas clases heredadas Barco y Velero de acuático y Avión y Helicóptero de aéreo.



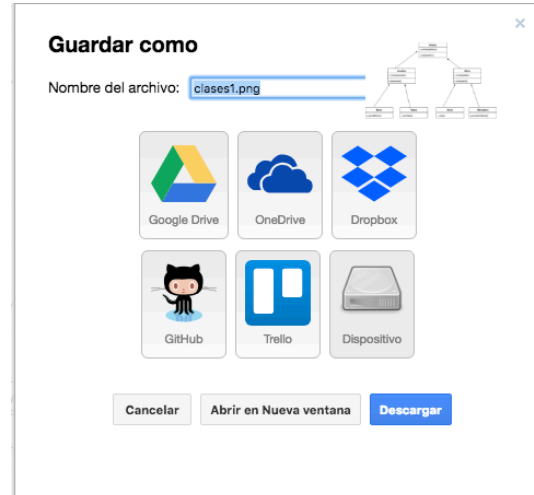
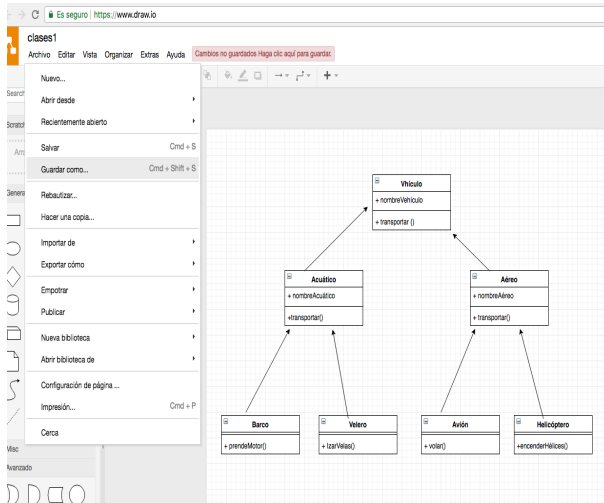


### 5. Conectar las clases

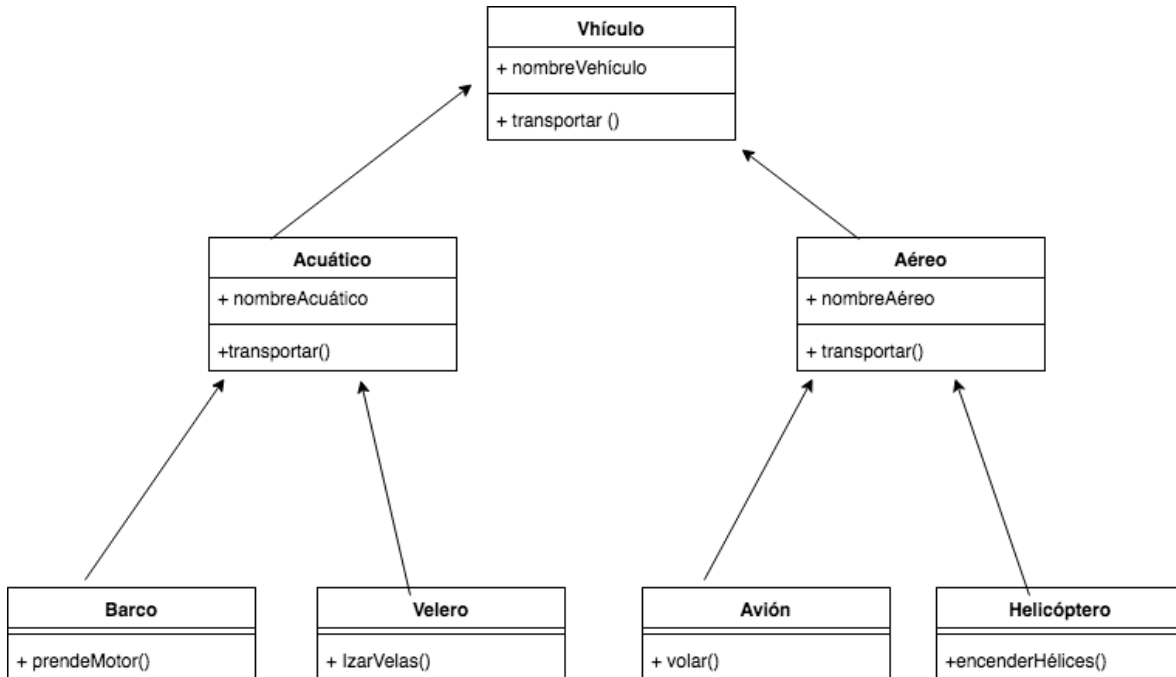


6.- Guardar y exportar en formato de imagen y agregarlo en la ubicación que desees archivarlo





### 7. Descargar la imagen.



### Síntesis

La jerarquía de clases es la mejor forma de hacer reciclaje de componentes, si tenemos clases en la colección que sirvan para la construcción de nuevos sistemas, bastará con hacer uso de ellas, esto reduce tiempo de desarrollo. Los diagramas de clases que representan la jerarquía de clases muestran al alumno esas asociaciones que facilitan la abstracción de este





mecanismo. Cada clase derivada hereda los elementos de su clase base y así esa clase base hereda elementos de su clase base, y así se hace sucesivante.

### **Preguntas de Repaso**

1. Qué significa el + en un diagrama de clases.
2. Que datos se anotan en la primera parte de la clase.
3. Que datos se anotan en la segunda parte de la clase.
4. En el ejercicio en la clase velero que atributos y métodos contendrá la clase velero, tomando en cuenta que es una subclase de acuático y asu vez de vehículo.
5. En el ejercicio en la clase Avión que atributos y métodos contendrá la clase aéreo, tomando en cuenta que es una subclase de acuático y asu vez de vehículo.

### **Referencias**

Deitel H. M. y P.J. (2008). Como programas C++. 6ª. Edición. México. Pearson Educación.

Kappel, G., Brandsteidl, M., Huemer, C., y Seidl, M. (2012). UML @ Classroom: una introducción al modelado orientado a objetos, Springer.

Kendall K. y Kendall J. (2011). Análisis y diseño de sistemas. 8ª edición. México. Pearson Educación.

Fowlr, M y Kendall, S.(1997). Uml gota a gota. Pearson Edicación, Massachisetts, Estados Unidos.

Larman, C. (2003) UML y patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado. México. Pearson Educación

Pantaleón G., Lis Rinuado, L. (2014)Ingeniería de software, Alfaomega

