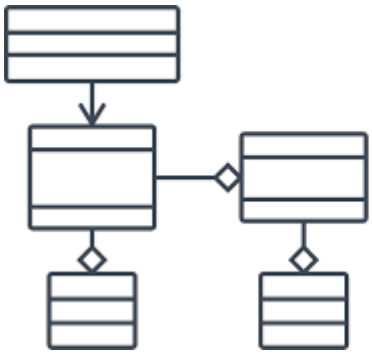
SESIÓN 1

Antes de empezar conocer este nuevo tema, conozcamos de donde viene

Diagrama UML

El Lenguaje Unificado de Modelado (UML), es el que permite a modelar los sistemas de software más conocido y utilizado en las grandes empresas y programadores. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

Cada una de las diversas gráficas representan una actividad y perspectiva del sistema, donde se le conoce como *módulos,* el mismo se encarga de describir lo que se supone hará el sistema.



A continuación verás los diversos diagramas que UML posee, y una definición corta:

	Diagrama de Clases	Diagrama de Objetos	Diagrama de Caso de Uso	Diagrama de Estado
Definición / Diagrama		Los diagramas de objetos describen la estructura estática de un sistema en un momento particular y son usados para probar la precisión de los diagramas de clases	Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario	En cualquier momento, un objeto se encuentra en un estado particular, la luz está encendida o apagada, el auto en movimiento o detenido, la persona leyendo o

				cantando, etc. El diagrama de estados UML captura esa pequeña realidad.

.

	Diagrama de Secuencia	Diagrama de Actividades	Diagrama de Colaboración	Diagrama de Componentes	Diagrama de Distribución
Definición / Diagrama	El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos.	Ilustra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de actividad en actividad. Una actividad representa una operación en alguna clase del sistema y que resulta en un cambio en el estado del sistema	El diagrama de colaboraciones describe las interacciones entre los objetos en términos de mensajes secuenciados.	Un diagrama de componentes describe la organización de los componentes físicos de un sistema.	El diagrama de distribución UML muestra la arquitectura física de un sistema informático.

Si se pudo observar el Diagrama de Clases no tenia nada definido, ya que ese el UML con el que comenzaremos a trabajar en esta sesión.

Diagrama de Clase

Es un diagrama que representa de manera estática un sistema, es decir, es la visión del sistema ejemplificado en un diagrama donde se puede ver las diversas relaciones que tiene un módulo con otro. Este diagrama solo funciona cuando se trabaja con la Programación Orientada a Objeto (este tema lo veremos más adelante), pero en el caso de la Programación Estructura para lenguajes como C y C++ se usa el Diagrama Modular, estos 2 diagramas tienen gran similitud entre ellos.

El Diagrama de Clases como su nombre los dice está compuesta por diversas estructuras llamadas *clases*, al llamarse diagrama hace referencia a un conjunto, por ende podemos decir que Diagrama de Clase puede ser traducido a Conjuntos de Clases; adicionalmente posee relacione y Objetos dentro de su estructura.

Este tipo de diagrama se utiliza para definir cómo va orientado el sistema, para estructurar cuales son los módulos y submódulos que poseerá, además de las características y acciones que realizará el usuario durante su recorrido.

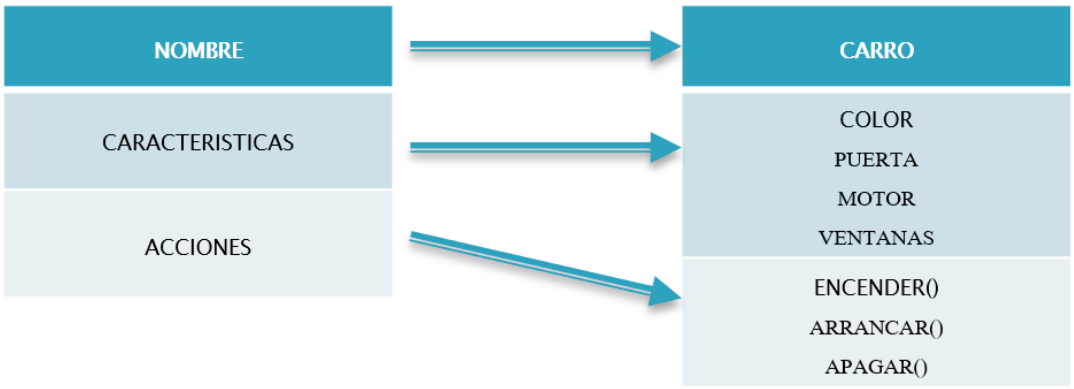
Clases

La clase es una representación donde se **describen un conjunto de objetos con características y acciones**, esto hace que se modelice la conceptualización del sistema.

Se representa como un rectángulo verticalmente dividido en 3 partes, donde la parte superior se enfoca en el nombre de dicha clase por ejemplo: Casa, Avión, Registro, Juego; por otro lado la parte del medio hace referencia a todas las características y atributos que posee esa clase, por ejemplo: Estatura, peso, color, nombre, código; y la parte inferior se

caracteriza por definir todos los métodos y acciones que realiza dicha clase, se puede definir como: Hablar, Encender, Correr, Dormir, Volar, entre otras,

Se representa gráficamente de la siguiente manera:

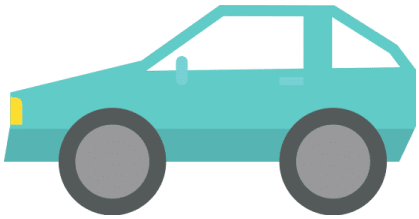


Atributos

Esto describe las **características propias de la clase u objeto**, la mismas son representaciones únicas del objeto que se le asigne. Estas características están definidas por 3 tipos, pero en esta sesión solamente veremos 2 de ellos.

- **Públicos**: son los representados por el signo de mas **(+)** antes de los atributos, el mismo posee una característica que lo hace que pueda ser usado en todas las clases que tenga alguna relación.
- **Privados:** Son representados por el signo de menos **(-)** antes de los atributos. el mismo posee una característica que lo hace que pueda ser usado en la clase en la que se encuentra, son datos privados que otra clase no tiene acceso.

Podemos definir las características tanto un objeto inanimado como un objeto que tiene vida, en este ejemplo veremos las características de una persona y un carro.

	**Características**
	Ojos
	Manos
	Edad
	Color
	Estatura
	**Características**
	Marca
	Modelo
	Año
	Puertas
	Motor

Métodos

Esto describe las acciones o comportamientos que realiza la clase u objeto, la mismas son representaciones únicas del objeto que se le asigne, esto indica que funcionalidades tiene dentro de ella. Estas acciones están definidas por 3 tipos, pero en esta sesión solamente veremos 2 de ellos.

- **Públicos**: son los representados por el signo de mas **(+)** antes de los atributos, el mismo posee una característica que lo hace que pueda ser usado en todas las clases que tenga alguna relación.

- **Privados:** Son representados por el signo de menos **(-)** antes de los atributos. el mismo posee una característica que lo hace que pueda ser usado en la clase en la que se encuentra, son datos privados que otra clase no tiene acceso.

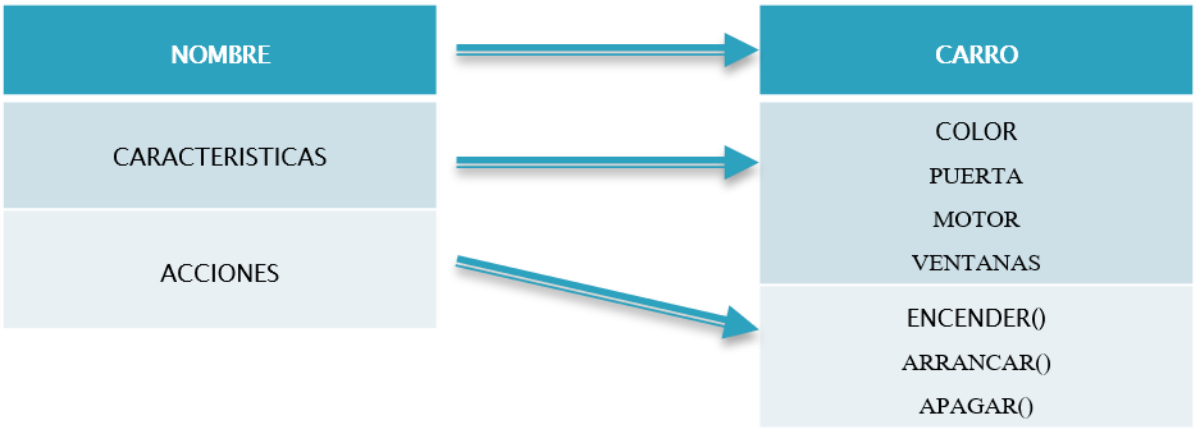
Nota: al terminar de escribir cada una de las acciones o métodos se debe colocar un par de paréntesis **( )** al final.

Manteniendo el ejemplo que se presentaron en los atributos, veremos como podemos ver las diferentes acciones que hace cada uno de ellos

	**Acciones**
	Despertar ()
	Comer ()
	Caminar ()
	Correr ()
	Dormir ()
	**Acciones**
	Encender ()
	Apagar ()
	Acelerar ()
	Retroceder ()
	Avanzar ()

En los ejemplos presentados de los atributos y métodos podemos observar que uno hace referencia a las **características** que contiene ese objeto, y en el caso de los métodos son las **acciones** que realiza esos objetos, que estos culminan cada uno con un par de paréntesis.

Realizar: *Para saber si comprendiste esta parte de la sesión vas a realizar las siguientes clases, es decir, vas a dibujar en tu cuaderno de informática 1 el rectángulo definiendo los Atributos y métodos, haciendo un mínimo 4 cada uno.* ***Clases:*** *Avión, Restaurante, León, Serpiente, Rinoceronte, Submarino*



SEGUIMOS!

Hasta ahora solo hemos visto una (1) clase, y lo mencionado al principio es la representación de varias clases. Por ende te presentamos:

Relaciones.

Esto permite unir las clases uno con otra, esto solo funciona al unir 2 clases al mismo tiempo. Las mismas indican que tipo de relación tiene cada una de las clases y como van

orientado dentro de la estructura del sistema. Las relaciones son las que permite hacer que varias clases por separado se conviertan en un *Diagrama de clases.*

Al momento de conformar las relaciones de estas clases, se tiene que tomar en cuenta los diferentes tipos que existen, donde cada uno tiene una utilidad particular.

Tipos de Relaciones:

- ***Generalización o Herencia:*** Es una relación que permite que una *Clase Hijo* **herede** la información de otra clase llámese *Clase Padre*. Esto permite que la Clase hijo pueda tener los atributos y métodos de las Clase Padre, es lo mismo cuando te dicen: "*Tienes los ojos de tu madre", "Te pareces a tu papá",* esto viene realizado por una relación de herencia, que a nivel científico llamamos ADN.

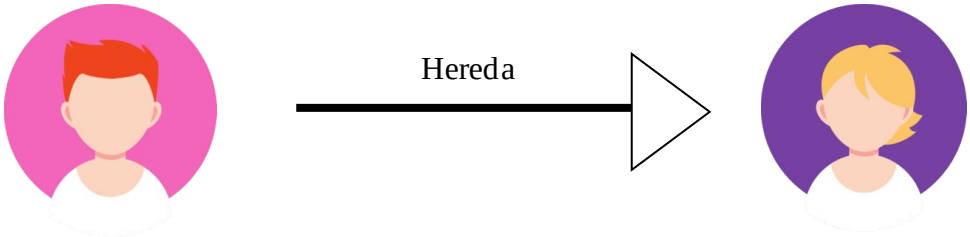
  Esta relación se representa de la siguiente manera:

  La flecha que apunta a la *Clase Padre* y la cola a punta a la *Clase Hijo*

  Ejemplo.

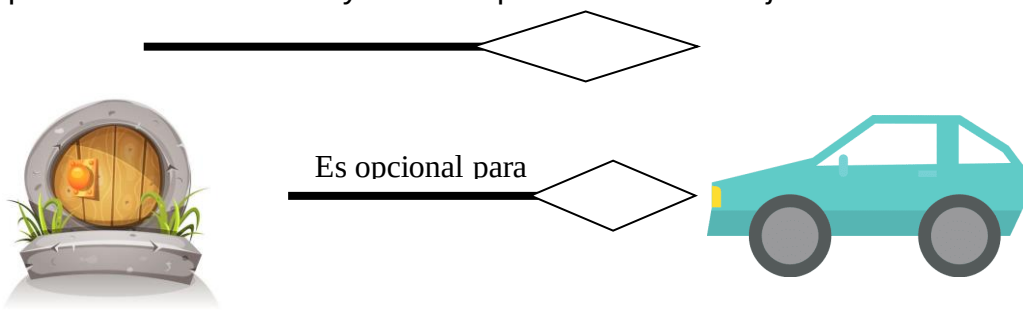
  Cuando un hijo Hereda toda la información en este caso genes de la madre.

- ***Agregación:*** Esta relación se usa cuando algo es *opcional*, es decir, que no es obligatorio para esa clase, esta relación es dinámica, ya que permite que se pueda borrar y así no afectar la información que se tiene almacena o que se

envía de una clase a otra. Además permite que una *Clase Hijo* **reciba valores opcionales** de otra clase llámese *Clase Padre.*

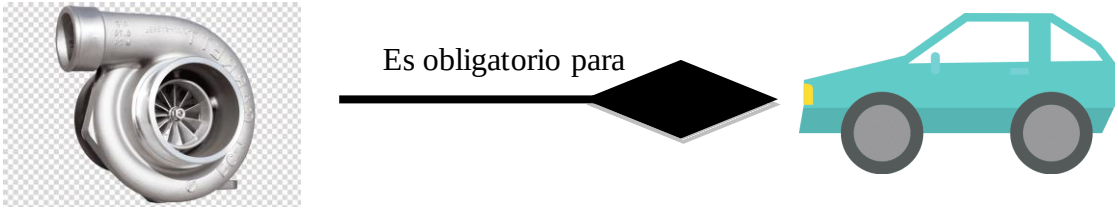
- Esta relación se denota de con rombo **blanco**, no posee color alguno. La flecha que apunta a la *Clase Padre* y la cola a punta a la *Clase Hijo.*



Un carro puede funcionar sin una puerta, por eso una relación opcional, que al eliminarse todavía el objeto puede funcionar.

- ***Composición:*** Esta relación se usa cuando algo es *obligatorio*, es decir, que sí es obligatorio para esa clase, esta relación es estática, ya que si se borra la información la misma puede se elimina en ambas clases. Además permite que una *Clase Hijo* **reciba valores obligatorios** de otra clase llámese *Clase Padre.*

Esta relación se denota de con rombo **negro**. La flecha que apunta a la *Clase Padre* y la cola a punta a la *Clase Hijo.*



El motor es obligatorio para que el carro funcione. Cuando es obligatorio es algo que necesario para que funciones.

Reto de la semana

Debe hacer en una presentación de Power Point las siguientes clases, además deben colocarle una (1) de las relaciones. Tiempo de entrega máxima **sábado 25/4.**

Clases: Avión, Mamífero, Barcos, Humano, Moto, León.