



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES



Programación Orientada a Objetos

Cómo trabajamos hasta ahora?

- Descomponíamos el problema en subproblemas hasta llegar a acciones simples de codificar
- Los programas se descomponen en llamadas a funciones

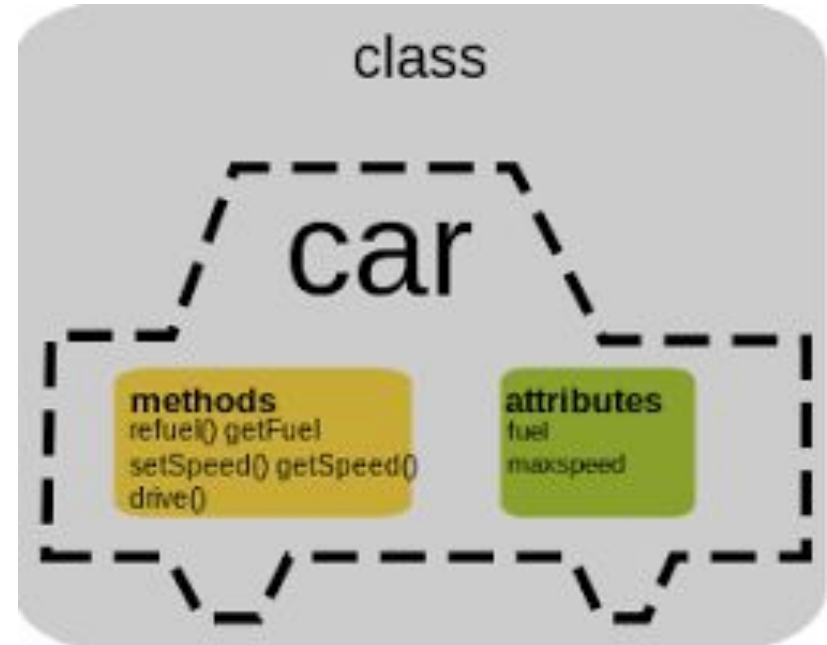
En Programación Orientada a Objetos

- Descomponemos el problema en objetos, intentando simular un escenario mucho más real
- Permite modelar más cercano al mundo real
- Un objeto es la representación de un concepto que tiene un comportamiento en particular, que intentamos modelar en la aplicación

POO

Qué es un objeto?

- Crear un objeto es como crear nuestro propio tipo de variable.
- Este objeto puede contener múltiples variables (atributos/propiedades) para almacenar diferentes tipos de datos.
- Además puede contener sus propias funciones (métodos) para manipular las propiedades del objeto

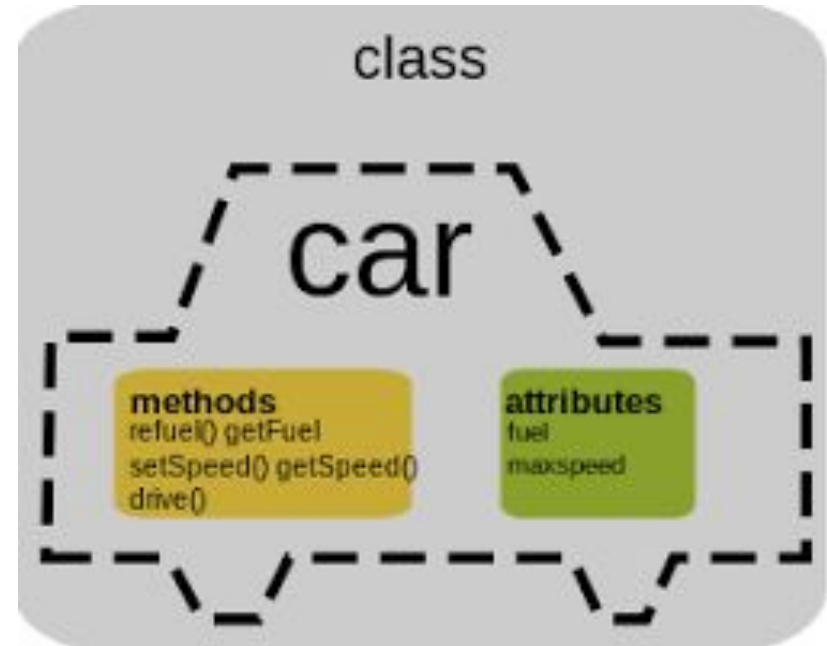


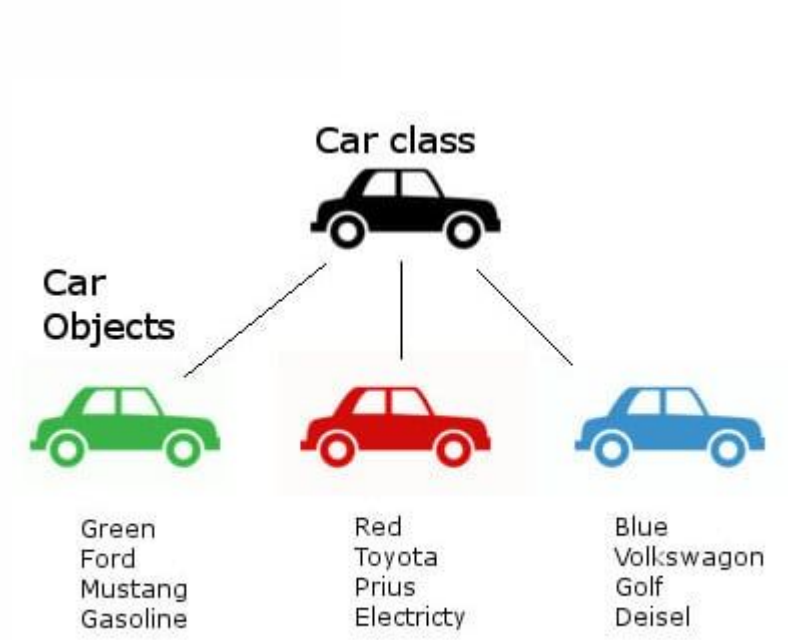
POO

Qué es un objeto?

Es un conjunto de:

- Propiedades (variables)
- Métodos (funciones)





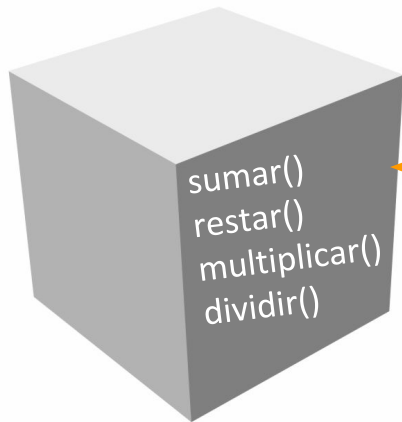
Plano -> Clase



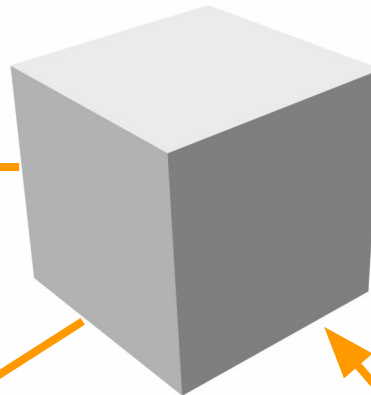
Cada casa -> Instancia



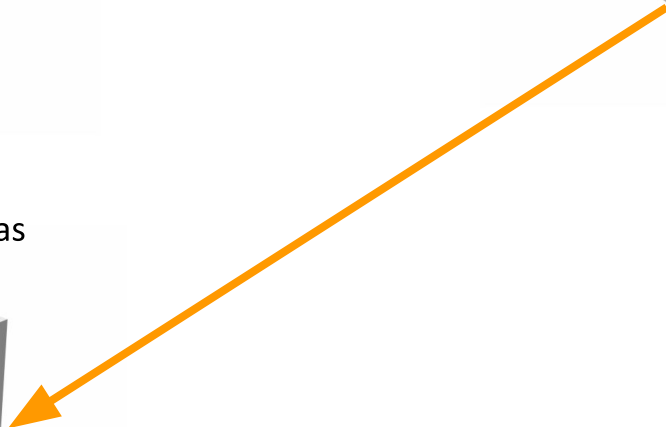
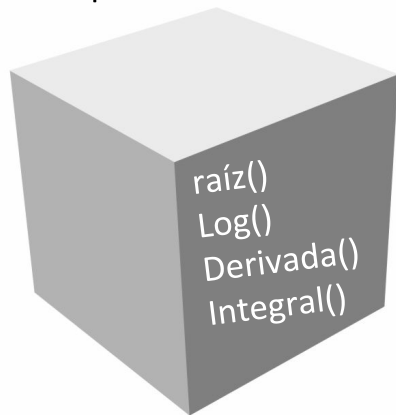
Operaciones



Calculadora



OperacionesAvanzadas



Crear y encastrar



El desafío está en cómo armar y “encastrar” las piezas

¿Qué es la POO?

Paradigma de programación que se basa en pensar la resolución de los problemas como una colección de objetos interrelacionados y que trabajan en conjunto.



Automóvil

Identidad: diferenciar y distinguir

Estado: atributos y sus valores en un instante de tiempo

Comportamiento: funcionalidad, operaciones

Objeto



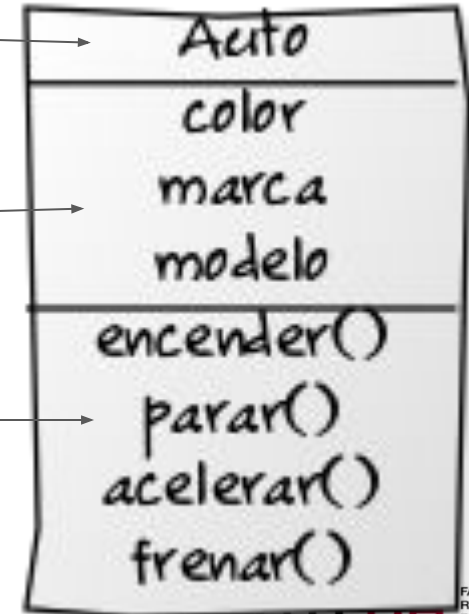
Identidad

Estado

Comportamiento

ATRIBUTOS

MÉTODOS



¿Qué es la POO?

ABSTRACCIÓN

- Nos permite concentrarnos en que tiene que hacer el objeto, y no como lo hace.
- Conocer "algo" en un alto nivel.

Abstracción significa representar la característica esencial sin detallar la implementación.

Ejemplo: todos los vehículos tienen algún tipo de frenado, aunque todos pueden implementarlo de manera diferente

¿Qué es la POO?

ENCAPSULACIÓN

- Se refiere a dar una solución a un problema sin necesidad que el consumidor entienda el problema, ni la forma de solucionarlo.
- Solo las propiedades y métodos que se deseen podrán ser accedidas por el consumidor ("puertas de entrada").

No es solo esconder la complejidad, también es exponer los accesos de una manera segura.

Clase



Abstracción y Generalización

Clase

Agrupación según las características en común

Vehículo

¿Qué es la POO?

CONVENCIONES GENERALES

- Las clases comienzan siempre con una letra mayúscula
- La separación de palabras es realizada por CamelCase (NombreDeMiClase)
- Una clase por archivo y el archivo con el nombre de la clase

PRÁCTICA

PRÁCTICA POO

Ejercicio 1

Si tenemos una clase Teléfono... ¿cuales podrían ser sus propiedades/características?

Tenés 1 minuto para escribir 3 propiedades que puede tener (Luego las ponemos en común)

Código en PHP

PHP - Definición de una clase

```
class UnaClase {  
    public $unaPropiedad;  
  
    public function unMetodo() {  
        echo "Hola desde el método";  
    }  
}
```

Creación de la clase

Para crear un objeto/instancia de la clase UnaClase, debemos utilizar la palabra new <nombre de la clase>();

Uso de la clase creada

```
$unaInstancia = new UnaClase();  
$unaInstancia->unaPropiedad = "Un valor";  
$unaInstancia->unMetodo();
```

Visibilidad de las propiedades y métodos

Si delante de una propiedad a un método usamos la palabra:

- **public:** el que instancia la clase puede acceder a esta propiedad/método
- **private:** el que instancia la clase NO puede acceder a esta propiedad método

Acceder a propiedades/métodos dentro de la clase

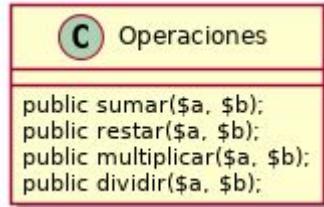
Se debe utilizar la palabra \$this-> No se debe incluir el \$ delante de la propiedad

```
class UnaClase {  
    public $unaPropiedad;  
  
    public function unMetodo() {  
        echo "Hola desde el método " . $this->unaPropiedad;  
    }  
  
    public function otroMetodo(){  
        $this->unMetodo();  
    }  
}
```

Crear la clase Tarea y definir las propiedades que debe tener

Agregar los métodos que consideres a la clase Tarea (recuerda que los métodos son acciones que se pueden hacer sobre la Tarea)

Crear la clase Operaciones con los métodos que se detallan a continuación



```
classDiagram
    class Operaciones {
        +sumar($a, $b)
        +restar($a, $b)
        +multiplicar($a, $b)
        +dividir($a, $b)
    }
```

Crear un código de ejemplo que utilice todos los métodos (sumar, restar, multiplicar, dividir)

Dado el código del siguiente slide:

- Implementarlo en el formulario de registración de usuario
- En caso que no haya completado todos los datos, retornar false
- Si completo todos los datos, retornar true

Práctica 04

```
class Formulario {  
    /**  
     * Valida que los datos cargados sean correctos  
     * - Si los datos son correctos => retorna true  
     * - Si los datos son incorrectos => retorna false  
     */  
    public function validar($post) {  
  
    }  
  
}
```

Práctica 04

```
$formulario = new Formulario();  
if ($formulario->validar($_POST)) {  
  
} else {  
  
}
```

Ejemplo de llamado



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES