



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Programador Web Avanzado

Clase N° 2: Programación orientada a objetos

Profesor: Ing. Leandro Rodolfo Gil Carrano

Email: leangilutn@gmail.com



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Clase N° 3: Programación orientada a objetos

- ¿Qué es la programación orientada a objetos?
- Definición de clase
- Objetos: instancias de una clase
- Constructor y destructor
- Operador \$this
- Propiedades y métodos estáticos
- Encapsulamiento en PHP5
- Herencia
- Un ejemplo practico

¿Qué es la programación orientada a objetos?

- Forma especial de programar
- Forma de programar mas cercana a la vida real
- Pensar la programación de forma distinta
- Programaremos pensando en objetos, propiedades y métodos
- **Todo lo debemos pensar como objeto y mensaje**

¿Qué es una clase?

La clase es utilizada para definir un objeto.

Dentro de la clase declaramos:

- Atributos (propiedades)
- Métodos (serán utilizados por el objeto)

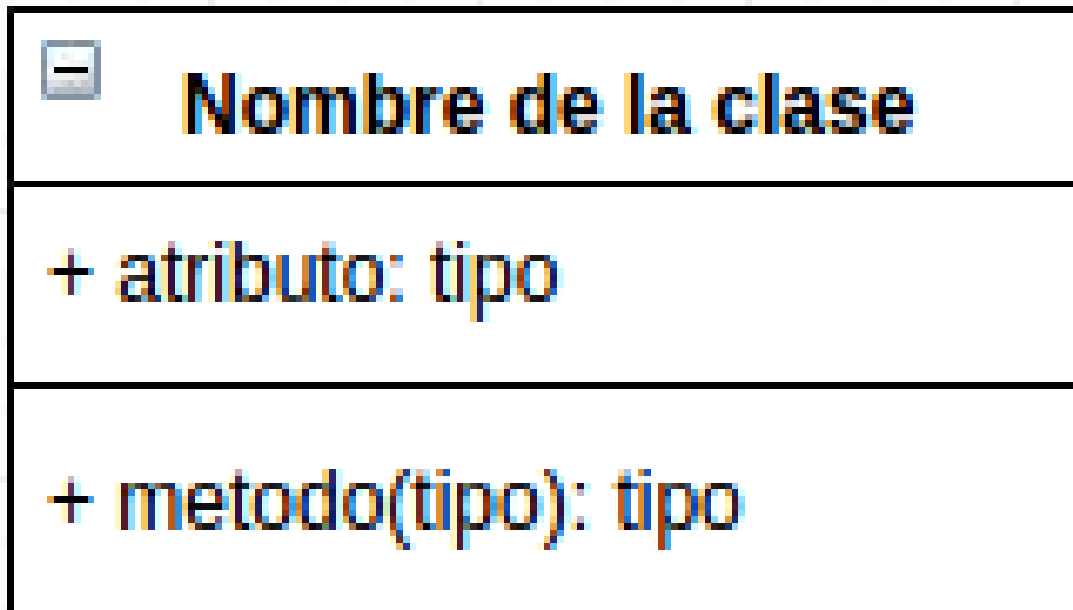
Entonces: una clase es el conjunto de declaraciones que van a definir los tipos de objetos que utilizaremos en el futuro.



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Diagrama de clase

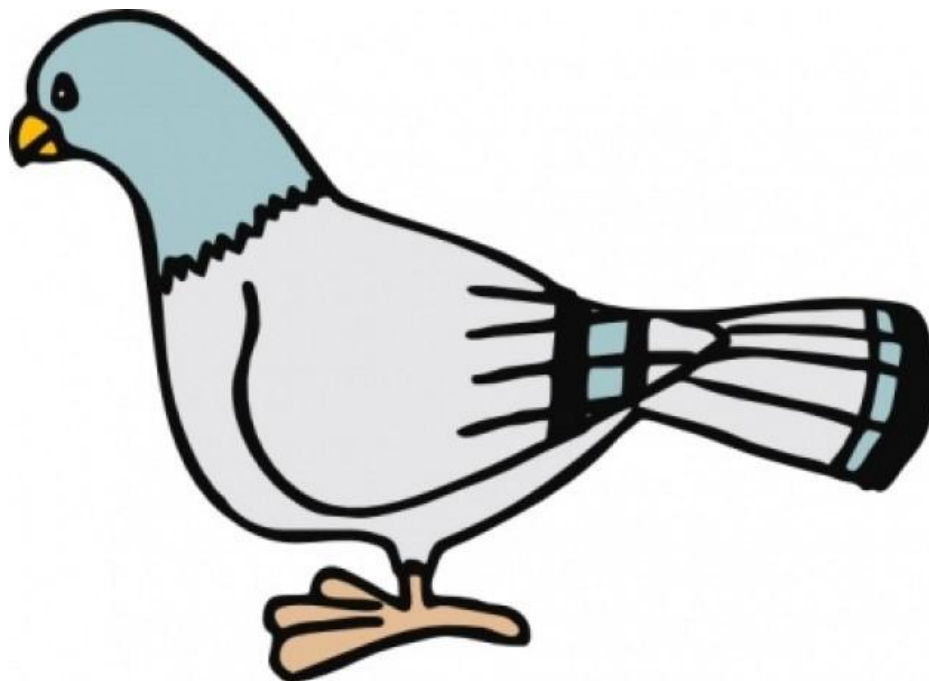




UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Te presento a nuestra amiga



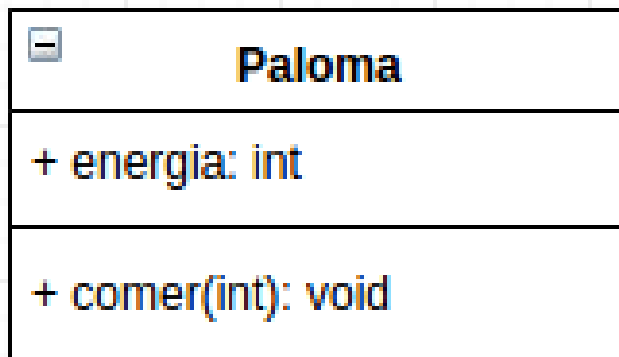
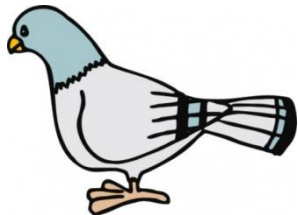
Pepita es nuestra paloma, la cual nos va ayudar a entender la
POO



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Ejercicio 1 - El diagrama de clases de pepita



Agregar al diagrama de clases , atributos:

- Días de vida
- Km recorridos

Métodos:

- Volar

Agregar 1 atributo y 1 método mas



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

¿Qué es una clase?

A continuación vemos un ejemplo de cómo declarar una clase:

```
"use strict";  
class Paloma{
```

Clase

```
    comer(cantidad){  
        |   this.energia += (cantidad*5);  
    }  
}
```

Método

Ejercicio:

- Codificar el diagrama de clases realizado.

Atributos de una clase

Los atributos son

- Características
- Cualidades
- Propiedades distintivas de cada clase.
- Contienen información sobre el objeto.
- Determinan la apariencia, estado y demás particularidades de la clase.

Varios objetos de una misma clase tendrán los mismos atributos pero con valores diferentes.

Métodos de una clase

Los métodos son como las funciones en los lenguajes estructurados, pero están definidos dentro de una clase y operan sobre los atributos de dicha clase.

Los métodos también son llamados las responsabilidades de la clase. **Para encontrar las responsabilidades de una clase hay que preguntarse qué puede hacer la clase.**

El objetivo de un método es ejecutar las actividades que tiene encomendada la clase a la cual pertenece.

Los atributos de un objeto se modifican mediante llamadas a sus métodos.

Constructor

- Ejecutado al momento de inicializar el objeto
- Utilizado para inicializar la clase
- Permite asignar valores a los atributos al instanciarse el objeto

```
class Paloma{  
    constructor(nombre){  
        this.nombre = nombre;  
        this.energia = 0;  
        this.dias_vida = 0;  
        this.km_recorridos = 0;  
    }  
}
```

Objetos: instancias de una clase

El objeto es la representación útil de la clase que lo define

```
var pepita = new Paloma();  
pepita.comer(10);  
console.log("Pepita, energia:", pepita.energia)
```

Operador **this**

A través del operador **this** podemos acceder a métodos y variables de la misma clase.

```
comer(cantidad){  
    |   this.energia += (cantidad*5);  
}
```

Ejercicio 2 - Desarrollemos a pepita

Pepita tiene:

- Energía acumulada
- Km_recorridos
- Días de vida.

Cuando come:

- Energía aumenta en 5 por cada unidad.

Cuando vuela:

- Energía decrementa en 3 por km recorrido.

Cuando duerme:

- Energía aumenta en 2 y aumenta 1 día de vida.

Propiedades y métodos estáticos

Pueden ser accesibles sin necesidad de instanciar la clase.

Debemos agregar **static** justo después del modificador de acceso en la declaración de dicho método o propiedad.

```
static comer(cantidad){  
    |   this.energia += (cantidad*5);  
}
```

```
Paloma.comer(20);
```

Herencia

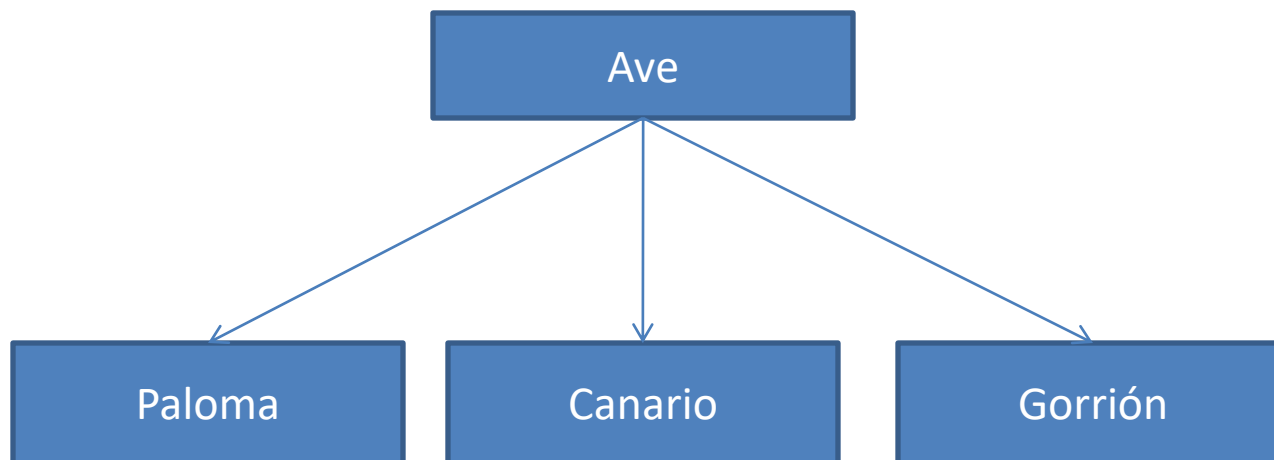
- Consiste en la definición de una clase a partir de otra
- Ej una clase B que hereda o extiende de la clase A
- La clase hija hereda métodos y atributos de la clase padre, pudiendo implementar sus métodos exclusivos



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Herencia



Redefinir métodos significa volver a codificarlos, es decir, volver a escribir su código para la clase extendida.



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Herencia

```
class Gorrion extends Ave{  
    constructor(nombre){  
        super(nombre);  
        this.decrementar_energia = 3;  
    }  
}
```

Herencia

Redefinir métodos o variables significa volver a codificarlos, es decir, volver a escribir su código para la clase extendida. Ejemplo el método comer esta redefinido

```
class Paloma extends Ave{  
  
    constructor(nombre){  
        super(nombre);  
        this.decrementar_energia = 2;  
    }  
    comer(cantidad){  
  
        this.energia += (cantidad*2);  
    }  
}
```

Herencia - Super

A través de **super** podemos hacer referencia a un método que este declarado en la clase “padre”.

```
class Canario extends Ave{  
  
    constructor(nombre){  
        super(nombre);  
        this.decrementar_energia = 2;  
    }  
    cantar(){  
        console.log("El canario canta");  
    }  
}
```

Herencia

- Comparten las acciones.: comer, volar, descansar
- El canario tiene una acción distinta que es cantar
- El gorrión cada vez que vuela decrementa su energía en 3 por km mientras que la paloma y el canario lo hacen en 2 por km.

Taller

- Crear una clase vehículos que tenga los siguientes atributos:
 - Atributos:
 - Combustible
 - Marca
 - Modelo
 - Cantidad de ruedas
 - Métodos:
 - Encender (decrementa M unidades el combustible)
 - Andar (decrementa M unidades por km recorrido)
 - Cargar (Incrementa el combustible N unidades)
 - Pintar (cambia el color del vehículo)

Taller

- Crear una clase auto que herede de vehículos:
 - Instanciar el objeto y definir las propiedades de vehículos
 - Agregar un atributo que indique si el auto es taxi (solo para la clase auto).
- Crear la clase moto que herede de vehículos:
 - Instanciar el objeto y definir las propiedades de vehículos
 - Agregar un método wheelie (decrementa la cantidad de combustible en 5)
- Crear la clase camioneta que herede de vehículos, definir 2 atributos y 2 métodos propios de camioneta que no sean compartidos por auto y moto.