

INTRO AL POO

Con Java

Charly Cimino

Intro al POO (con Java)

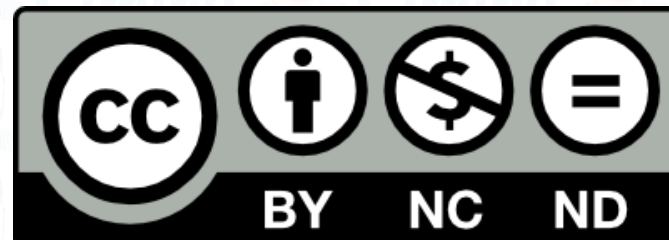
Charly Cimino

Este documento se encuentra bajo Licencia Creative Commons 4.0 Internacional (CC BY-NC-ND 4.0). Usted es libre para:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

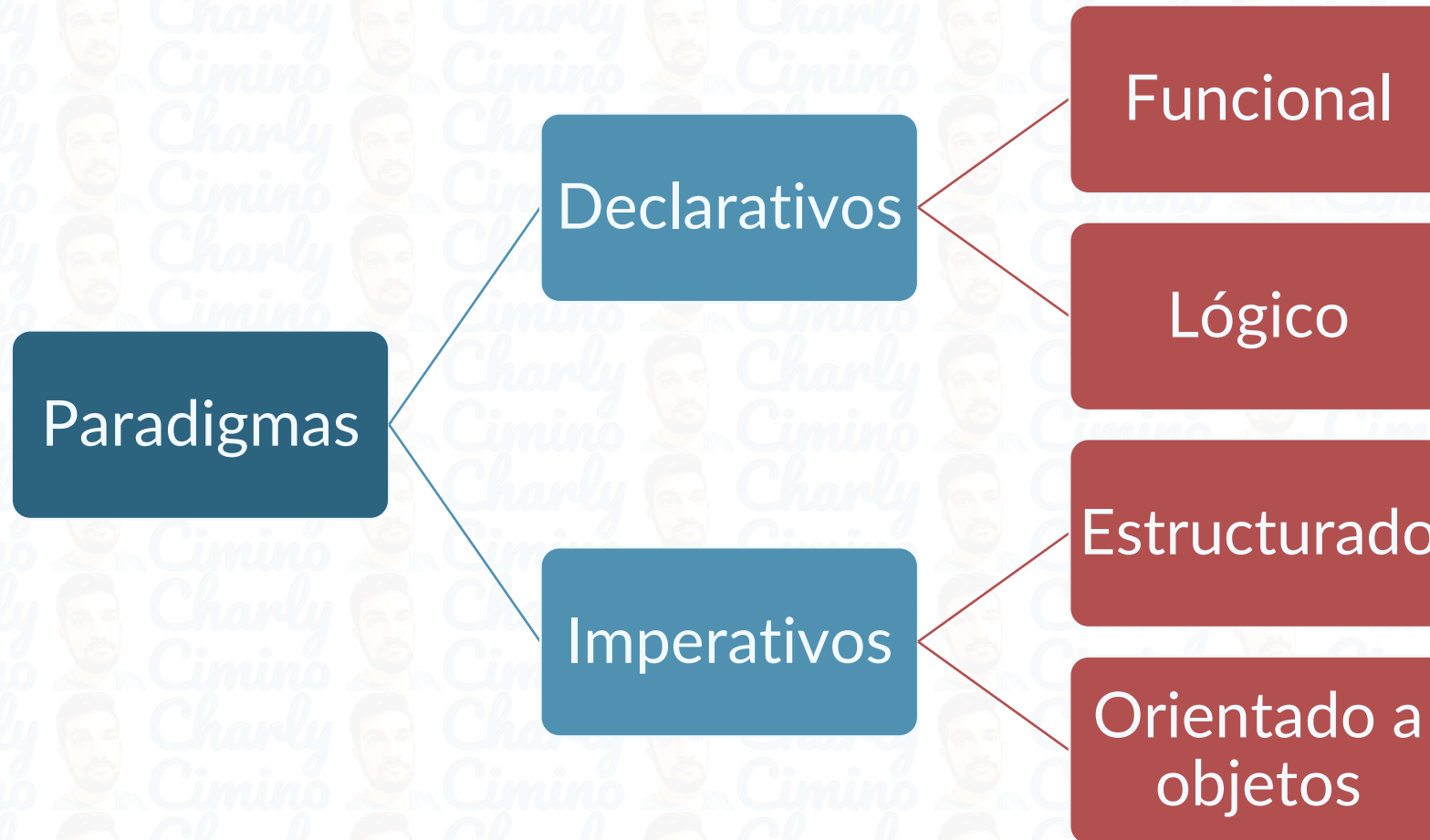
Bajo los siguientes términos:

- **Atribución** — Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.
- **No Comercial** — Usted no puede hacer uso del material con fines comerciales.
- **Sin Derivar** — Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted no podrá distribuir el material modificado.



Paradigma de programación

Conjunto de reglas y formas de razonamiento que conforman una filosofía para la producción de programas.

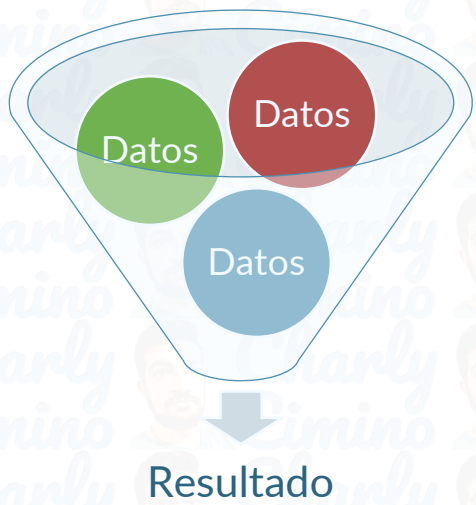


Paradigma orientado a objetos (POO)



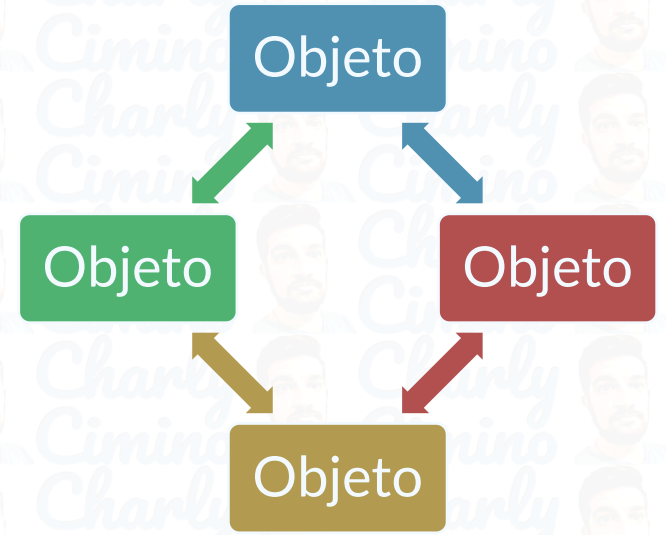
- Todo es un 'objeto'.
- La solución se basa en la interacción entre ellos.
- Se intenta modelar un subconjunto de la realidad.

Comparación entre paradigmas



Estructurado

El universo se compone de estructuras de datos y procedimientos capaces de procesarlos.



Orientado a objetos

El universo se compone de objetos con estado y comportamiento que interactúan entre sí.

Lenguajes de programación OO

Algunos de los lenguajes que permiten programar utilizando el paradigma orientado a objetos.

Smalltalk

C++

Java

JavaScript

C#

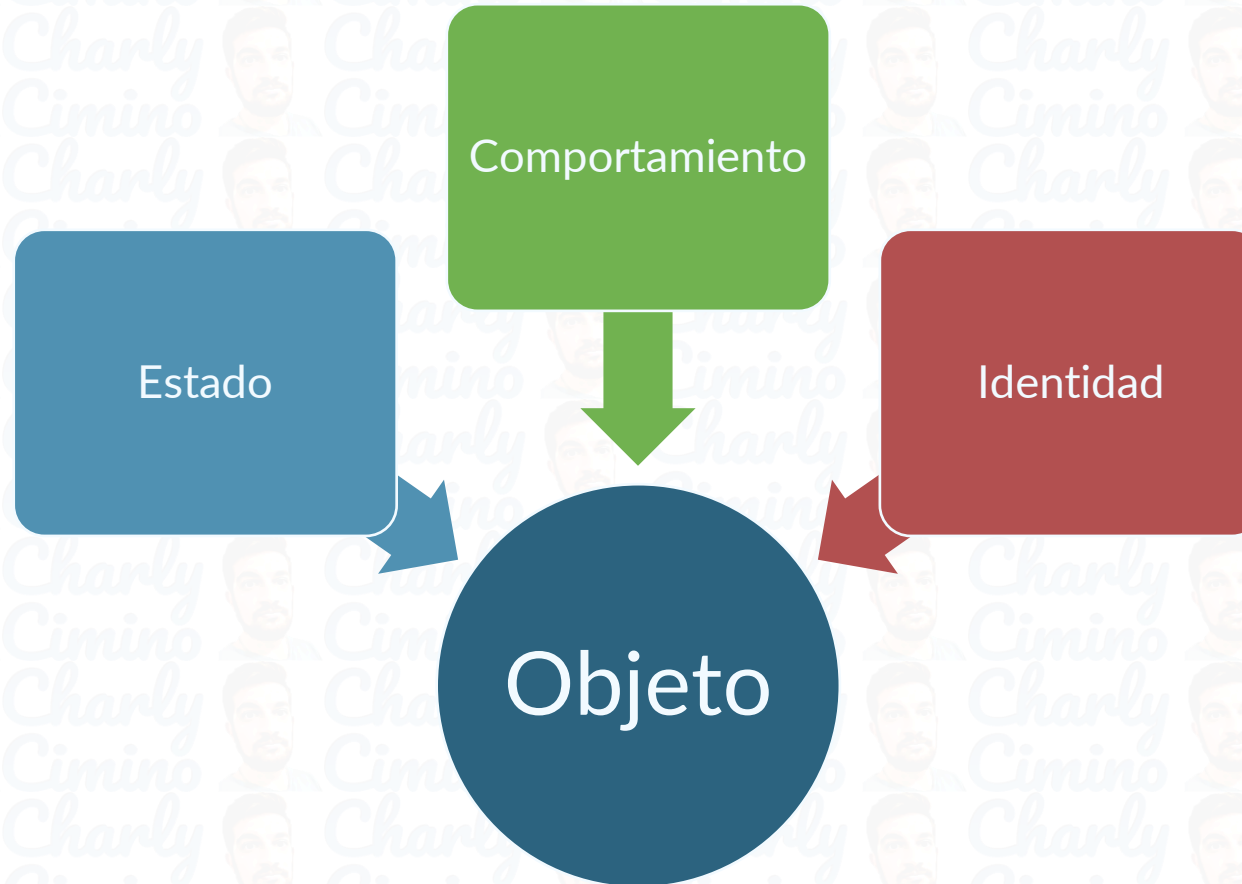
Python

PHP

Swift

Visual Basic

¿Qué es un objeto?



Estado de un objeto

Conjunto de valores de sus atributos en determinado instante.



Patente:	ABC123
Marca:	Acme
Kms:	3123
Color:	Verde

Atributos



Cambio de estado



Patente:	ABC123
Marca:	Acme
Kms:	3123
Color:	Azul

Valores

Comportamiento de un objeto

Operaciones que puede realizar a través de sus métodos.



- Acelerar
- Frenar
- Encender
- Cambiar de color



- Abrir cuenta
- Cerrar cuenta
- Alta de cliente
- Crear promoción

Identidad de un objeto

Propiedad que permite distinguirlo de otros.
Generalmente se trata del espacio que ocupa en la memoria



Auto@a1b2c3



Auto@d4e5f6

Dirección en memoria



Auto@67a8b9



Auto@c0da32

Objetos y clases

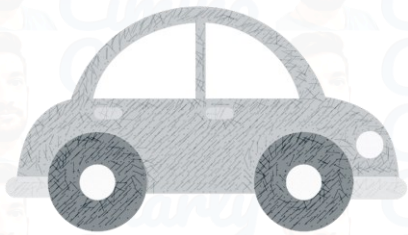
Objeto



Instancia de una clase

¿Qué es un clase?

Molde o plantilla que permite la creación de objetos de determinado tipo.



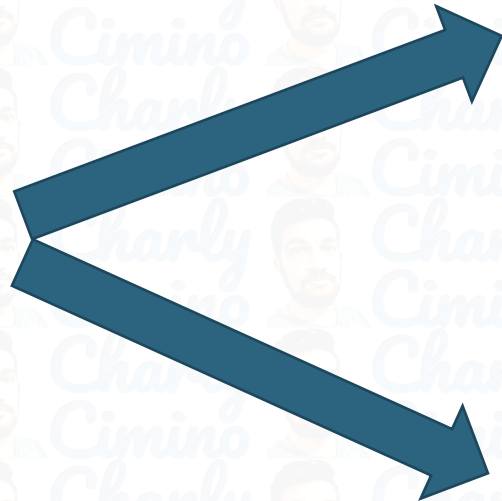
Clase Auto

Atributos

- Patente
- Marca
- Kms
- Color

Métodos

- Acelerar
- Frenar
- Encender
- Cambiar de color



Auto@a1b2c3



- Patente: ABC123
- Marca: Acme
- Kms: 1234
- Color: Verde
- Acelerar
- Frenar
- Encender
- Cambiar de color

Auto@d7e8f9



- Patente: XYZ456
- Marca: Foo
- Kms: 4233
- Color: Rojo
- Acelerar
- Frenar
- Encender
- Cambiar de color

Clases en Java

CuentaBancaria.java

```
public class CuentaBancaria {  
  
}
```

Nombre del archivo

Auto.java

```
public class Auto {  
  
}
```

Nombre de la clase

Fecha.java

```
public class Fecha {  
  
}
```

- Se recomienda tener un archivo .java por cada clase.
- Por convención, se nombran con la primera letra en mayúsculas.
- El nombre del archivo debe llamarse igual que la clase.

Fecha.java

```
public class Fecha {  
    int dia;  
    int mes;  
    int anio;  
}
```

Los atributos pueden ser de tipos primitivos u otras clases.

Auto.java

```
public class Auto {  
    String patente;  
    String marca;  
    int kms;  
    String color;  
}
```

CuentaBancaria.java

```
public class CuentaBancaria {  
    double saldo;  
    Persona titular;  
    Moneda moneda;  
    Fecha fechaApertura;  
}
```

Persona.java

```
public class Persona {  
    String DNI;  
    String apellido;  
    Fecha fechaNacimiento;  
}
```

Moneda.java

```
public class Moneda {  
    String nombre;  
    char simbolo;  
}
```


Definir métodos

Si bien el orden es irrelevante, por convención, los métodos se suelen listar a continuación de los atributos.

CuentaBancaria.java

```
public class CuentaBancaria {  
    double saldo;  
    Persona titular;  
    Moneda moneda;  
    Fecha fechaApertura;  
    double obtenerSaldo() {  
        return saldo;  
    }  
}
```

Auto.java

```
public class Auto {  
    String patente;  
    String marca;  
    int kms;  
    String color;  
    void cambiarColor(String nuevoColor) {  
        color = nuevoColor;  
    }  
}
```

Fecha.java

```
public class Fecha {  
    int dia;  
    int mes;  
    int anio;  
    String fechaComoCadena() {  
        return dia + "/" + mes + "/" + anio;  
    }  
}
```

Crear instancias

Para crear una instancia (objeto) de determinada clase, debe usarse la sentencia **new** seguida del nombre de la clase.

Auto.java

```
public class Auto {  
    String patente;  
    String marca;  
    int kms;  
    String color;  
    void cambiarColor(String nuevoColor) {  
        color = nuevoColor;  
    }  
}
```

Prueba.java

```
public class Prueba {  
    public static void main(String[] args) {  
        new Auto();  
    }  
}
```

new Auto();

← Crea un objeto de tipo Auto y devuelve su referencia en memoria (lo que le da identidad).

La referencia hacia el nuevo objeto se pierde.

Se necesita guardar en una variable.

Crear instancias

Si un objeto no es referenciado por ninguna variable/parámetro, Java lo borrará automáticamente de la memoria a través de su recolector de basura (*garbage collector*).

Auto.java

```
public class Auto {  
    String patente;  
    String marca;  
    int kms;  
    String color;  
    void cambiarColor(String nuevoColor) {  
        color = nuevoColor;  
    }  
}
```

Prueba.java

```
public class Prueba {  
    public static void main(String[] args) {  
        Auto unAuto = new Auto();  
    }  
}
```

unAuto es una variable que guarda la referencia hacia una instancia de auto.



Interactuar con un objeto

Para poder invocar a los miembros que componen un objeto (atributos y métodos), se debe usar el operador `.` (punto)

Auto.java

```
public class Auto {  
    String patente;  
    String marca;  
    int kms;  
    String color;  
    void cambiarColor(String nuevoColor) {  
        color = nuevoColor;  
    }  
}
```

Prueba.java

```
public class Prueba {  
    public static void main(String[] args) {  
        Auto unAuto = new Auto();  
        System.out.println(unAuto.color);  
        unAuto.cambiarColor("Rojo");  
    }  
}
```

Valores de los atributos

Cuando se crea un objeto, los atributos de éste tienen valores inicializados por Java automáticamente.

0

Numéricos

(byte, short, int, long, float, double, char)

false

Lógicos

(boolean)

null

Objetos

(String y demás...)

Cambio de estado

Un objeto puede cambiar de estado, idealmente a través de sus métodos.

Auto.java

```
public class Auto {  
    String patente;  
    String marca;  
    int kms;  
    String color;  
    void cambiarColor(String nuevoColor) {  
        color = nuevoColor;  
    }  
}
```

Prueba.java

```
public class Prueba {  
    public static void main(String[] args) {  
        Auto unAuto = new Auto();  
        System.out.println(unAuto.color); // Muestra null  
        unAuto.color = "Verde"; // No recomendado  
        System.out.println(unAuto.color); // Muestra Verde  
        unAuto.cambiarColor("Rojo"); // Forma correcta  
        System.out.println(unAuto.color); // Muestra Rojo  
    }  
}
```


FIN DE LA PRESENTACIÓN

Encontrá más como estas en mi [sitio web](#).