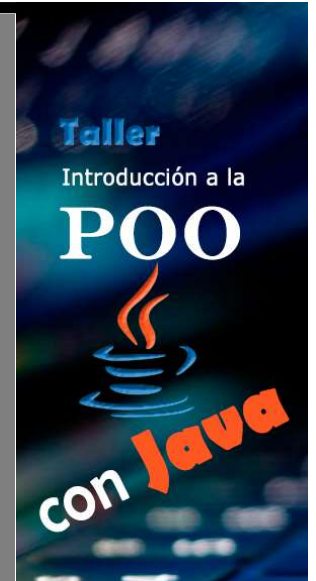
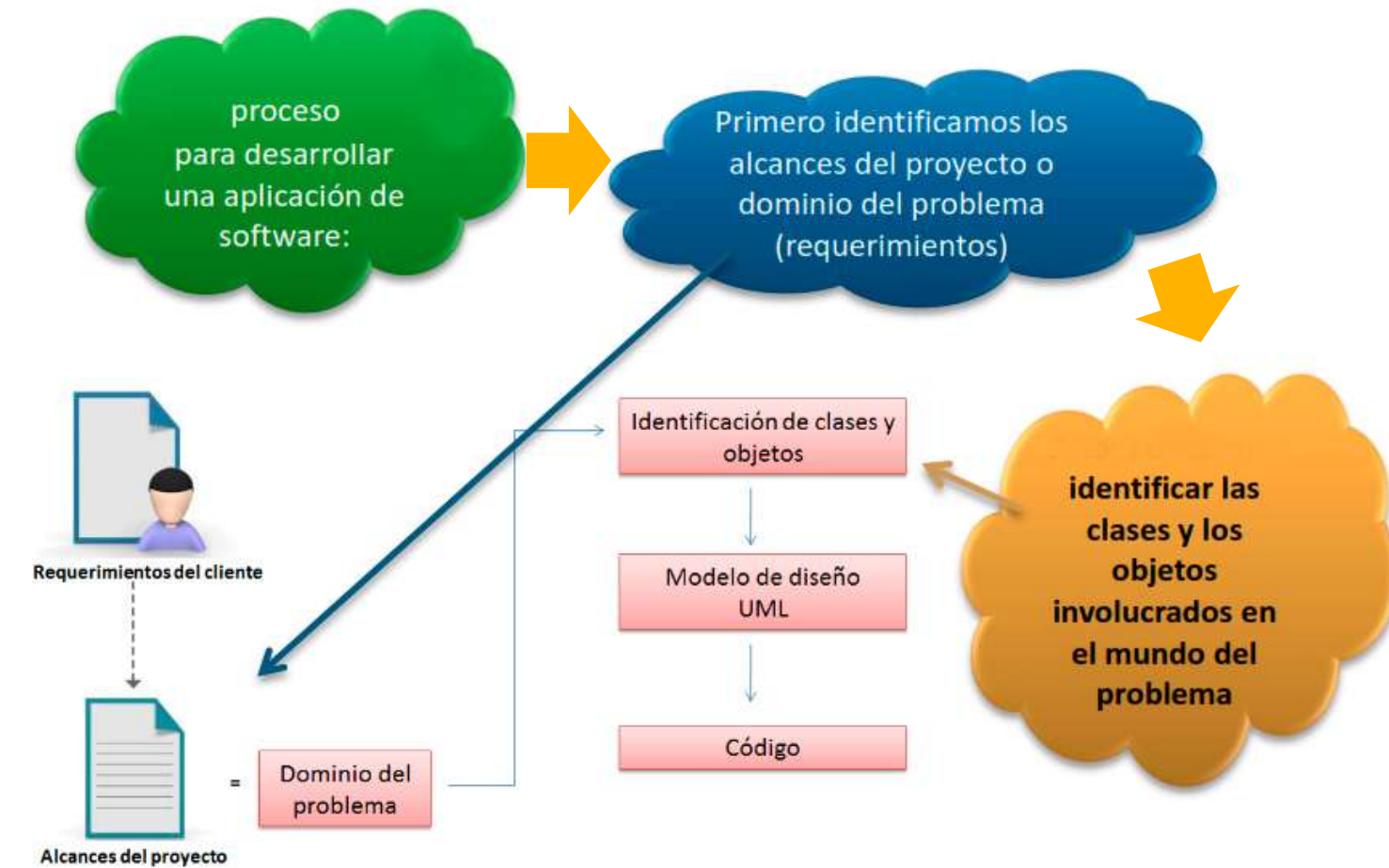


Pasos para aplicar POO



DESARROLLO DE UNA APLICACIÓN CON POO



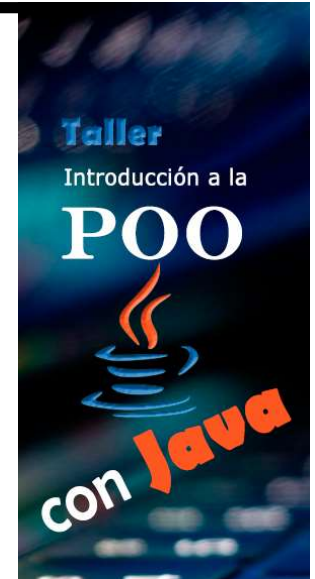
DESARROLLO DE UNA APLICACIÓN CON POO

- Consideremos el siguiente planteamiento de problema:

Un vehículo pertenece a una persona. El valor del vehículo se desvaloriza con el tiempo. Se necesita calcular el impuesto anual, que corresponde al 3% del valor del vehículo.

Los datos del vehículo son patente, marca, modelo, color, valor.

Los datos de una persona son dni, nombre, apellido.



Pasos a aplicar

- Identificación de datos, acciones y limitaciones
- Definición de clases
- Definición del método principal
- Desarrollo de diagrama UML de clases
- Desarrollo y prueba del código

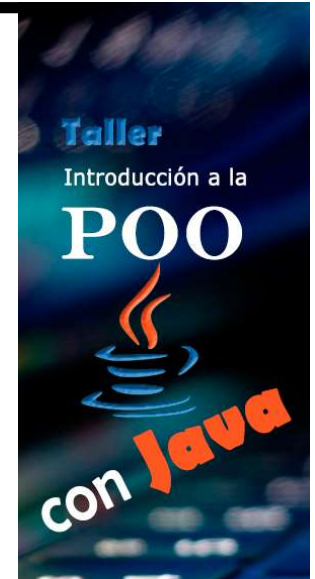
IDENTIFICACIÓN DE DATOS, ACCIONES Y LIMITACIONES

Un vehículo pertenece a una persona. El valor del vehículo se desvaloriza con el tiempo. Se necesita calcular el impuesto anual, que corresponde al 3% del valor del vehículo.

Los datos del vehículo son patente, marca, modelo, color, valor.

Los datos de una persona son dni, nombre, apellido.

- Datos de entrada: los datos del vehículo, los datos del dueño
- Datos de salida: impuesto anual igual al 3% del vehículo
- Acciones: calcular el impuesto anual
- Limitaciones: se hace el cálculo para un vehículo



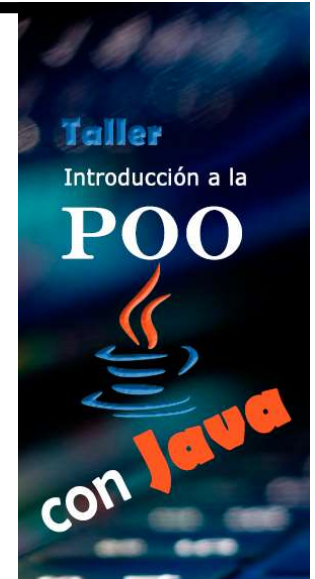
DEFINICIÓN DE CLASES

Un **vehículo** pertenece a una **persona**. El valor del vehículo se desvaloriza con el tiempo. Se necesita calcular el impuesto anual, que corresponde al 3% del valor del vehículo.

Los datos del vehículo son patente, marca, modelo, color, valor.

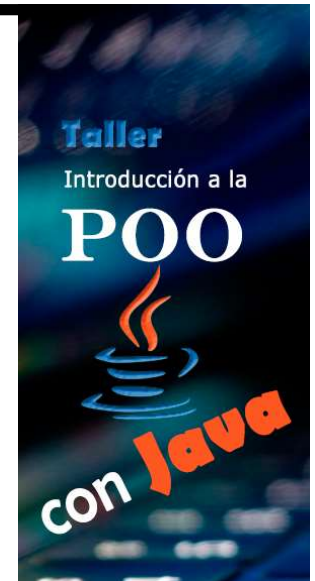
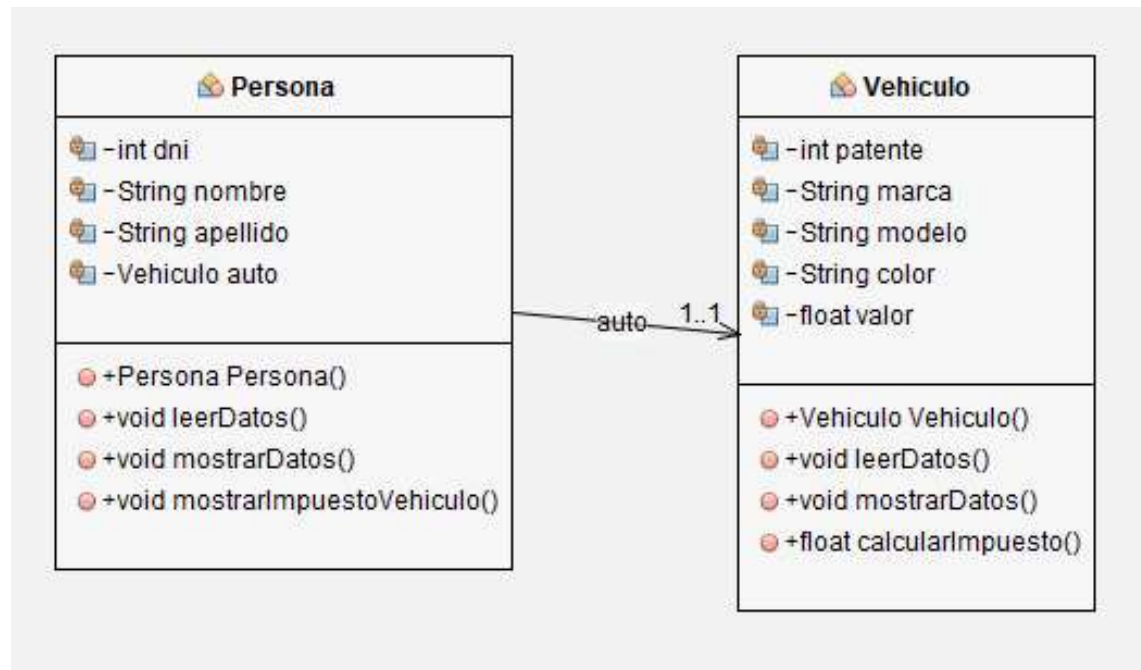
Los datos de una persona son dni, nombre, apellido.

- Sustantivos relevantes: vehículo, persona
- Relación de los sustantivos con datos y acciones:
 - Vehículo: tiene un dueño, patente, marca, modelo, color, valor y cálculo de impuesto
 - Persona: tiene dni, nombre, apellido, posee un vehículo
- Acciones requeridas para vehículo y persona:
 - Vehículo: constructor, leerDatos, mostrarDatos, calcularImpuesto
 - Persona: constructor, leerDatos, mostrarDatos, mostrarImpuestoVehiculo



DEFINICIÓN DE CLASES

Podemos crear el diagrama UML de clases a partir de las propiedades y comportamiento identificado



DEFINICIÓN DE LAS CLASES EN LENGUAJE JAVA

```
public class Persona {  
    private int dni;  
    private String nombre;  
    private String apellido;  
    private Vehiculo auto;  
  
    public Persona() {  
    }  
    public void leerDatos() {  
    }  
    public void mostrarDatos() {  
    }  
    public void mostrarImpuestoVehiculo() {  
    }  
}
```

```
public class Vehiculo {  
    private int patente;  
    private String marca;  
    private String modelo;  
    private String color;  
    private float valor;  
  
    public Vehiculo() {  
    }  
    public void leerDatos() {  
    }  
    public void mostrarDatos() {  
    }  
    public float calcularImpuesto() {  
        return 0;  
    }  
}
```

**Para completar
en el CUV**



- Completar la definición de las clases Persona y Vehículo.
- Identificar las operaciones a realizar en la clase Principal.
- Desarrollar las clases.



RESPONDE LA ENCUESTA EN EL CUV

