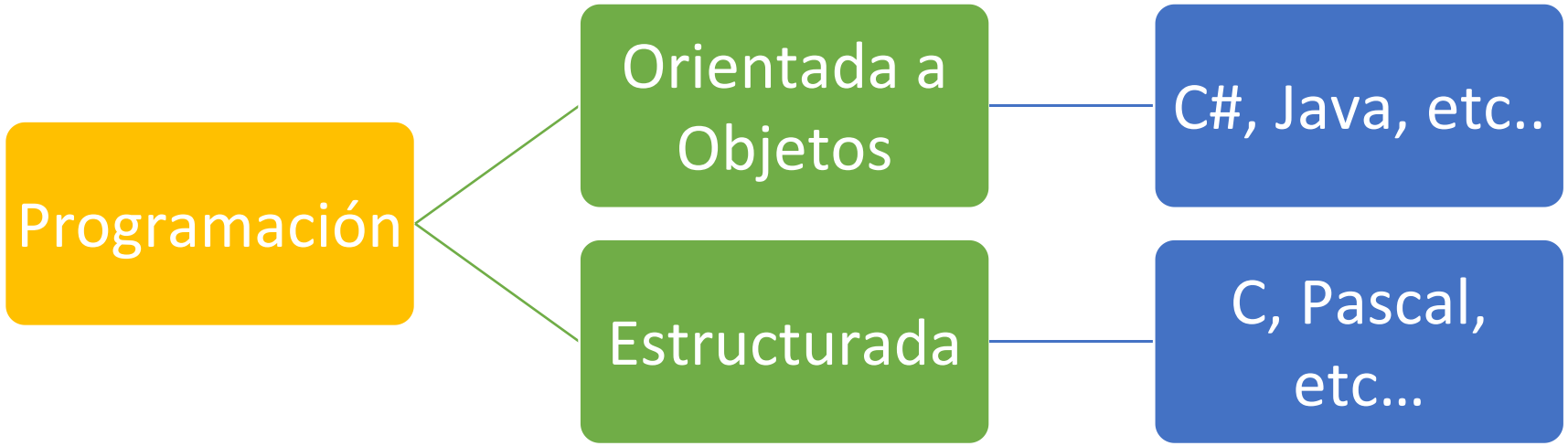
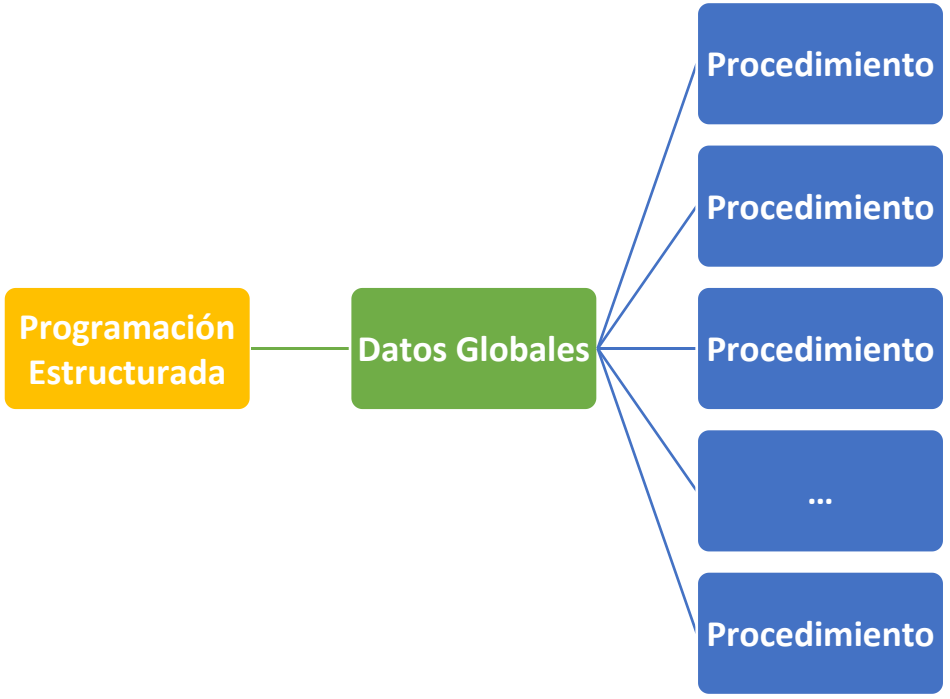
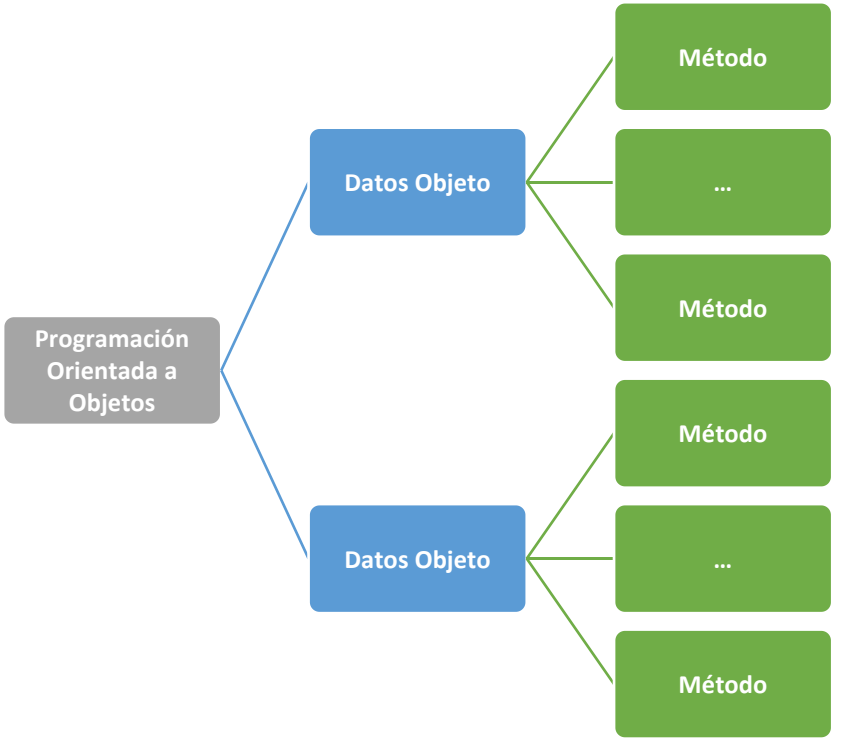
Programación Estructurada vs Programación Orientada a Objetos

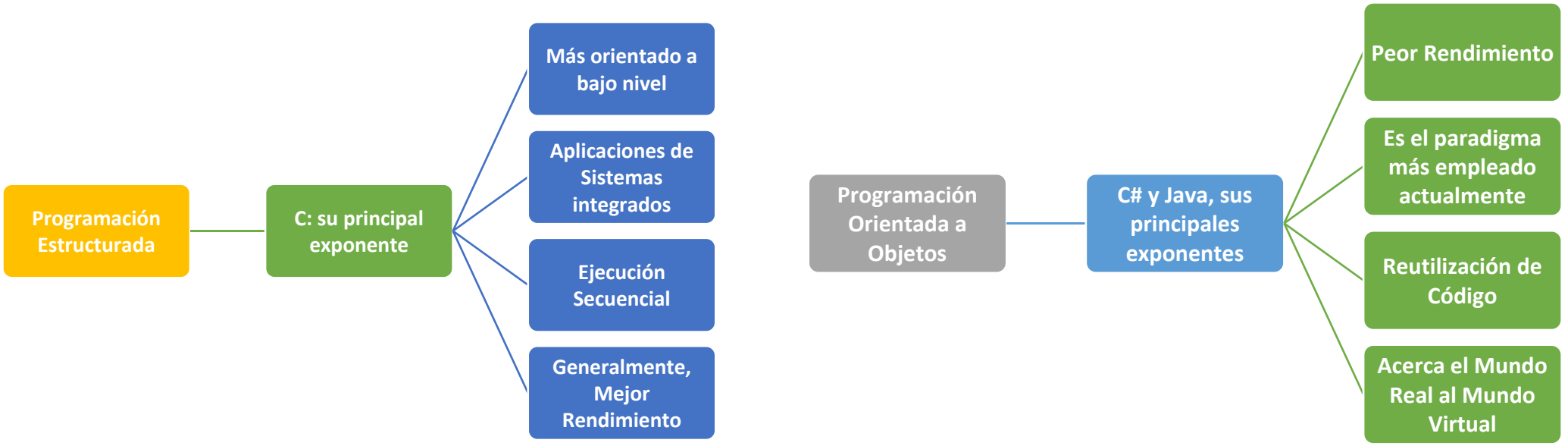


Programación Estructurada vs Programación Orientada a Objetos

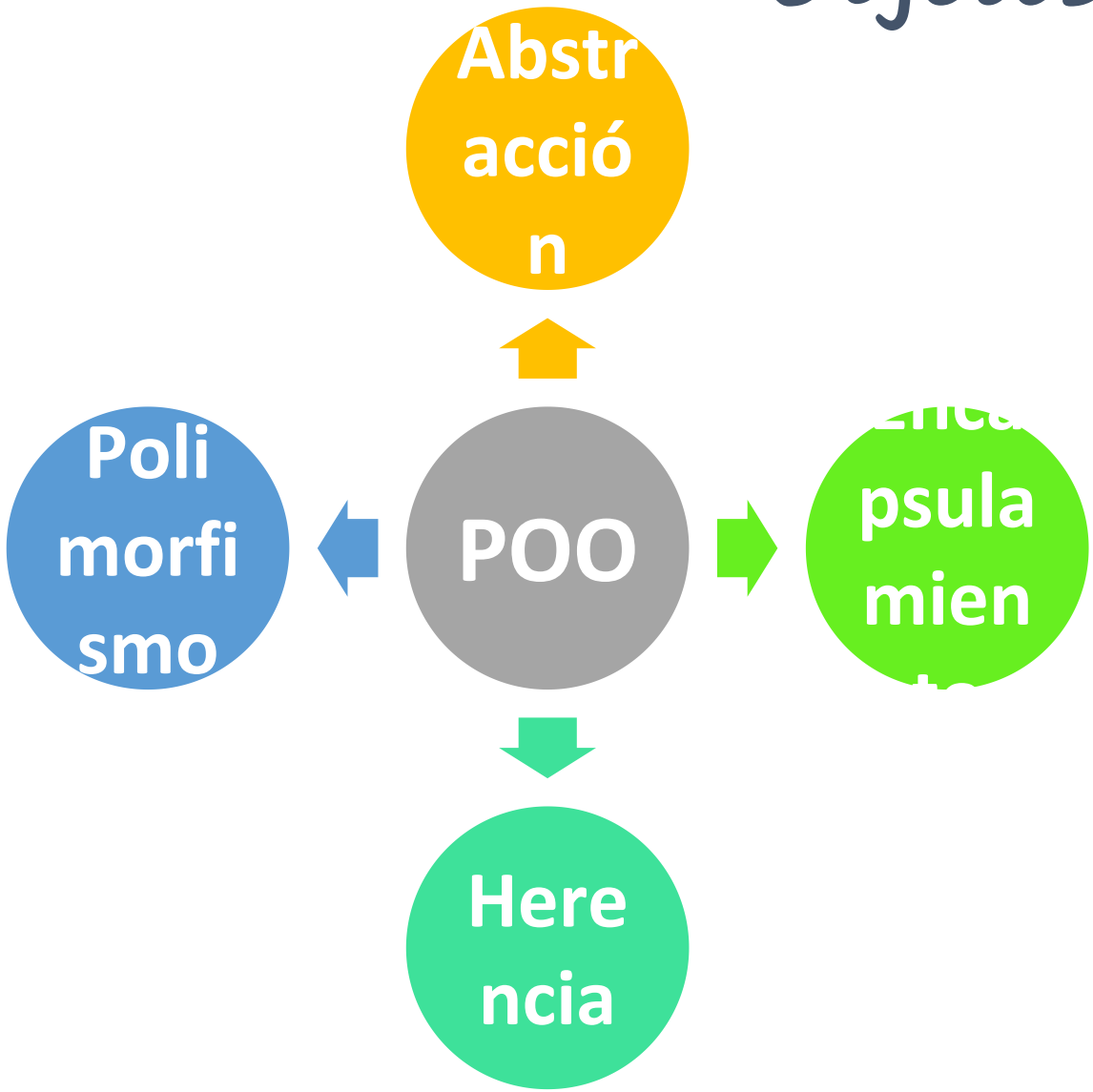




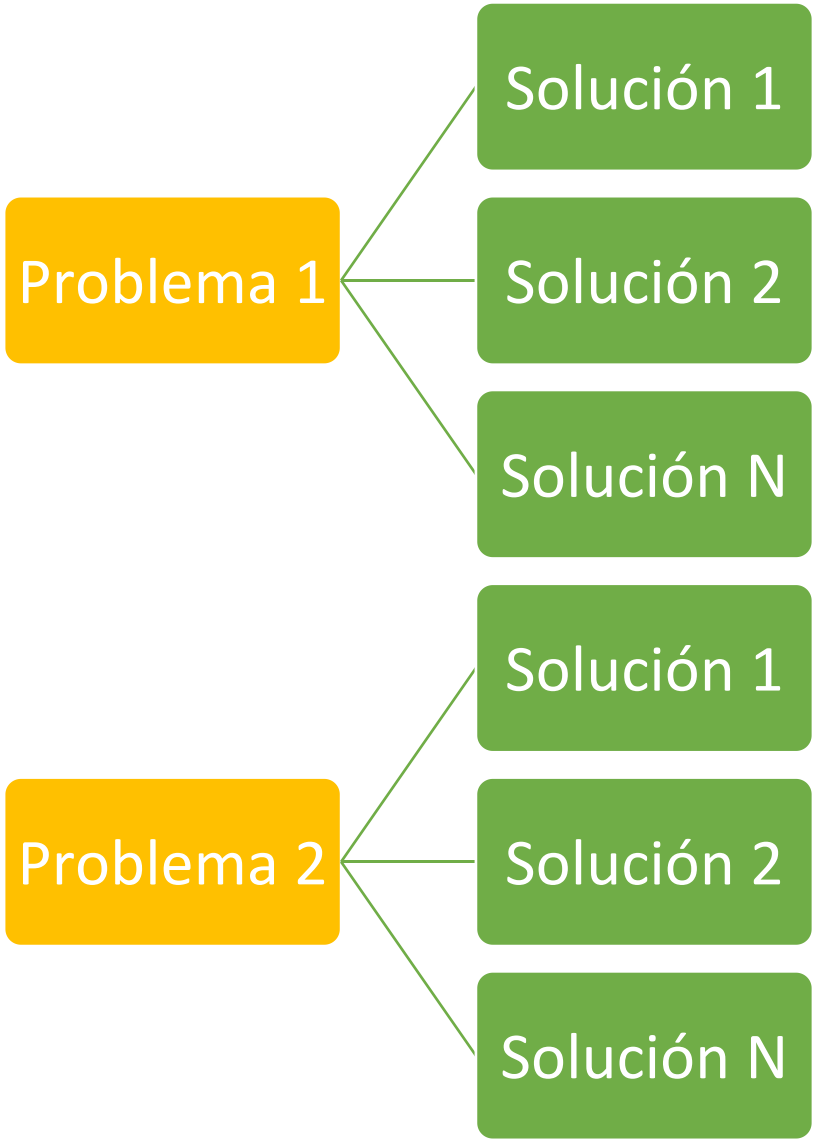
Programación Estructurada vs Programación Orientada a Objetos



Los 4 Pilares en la Programación Orientada a Objetos



Programación Orientada a Objetos - Abstracción



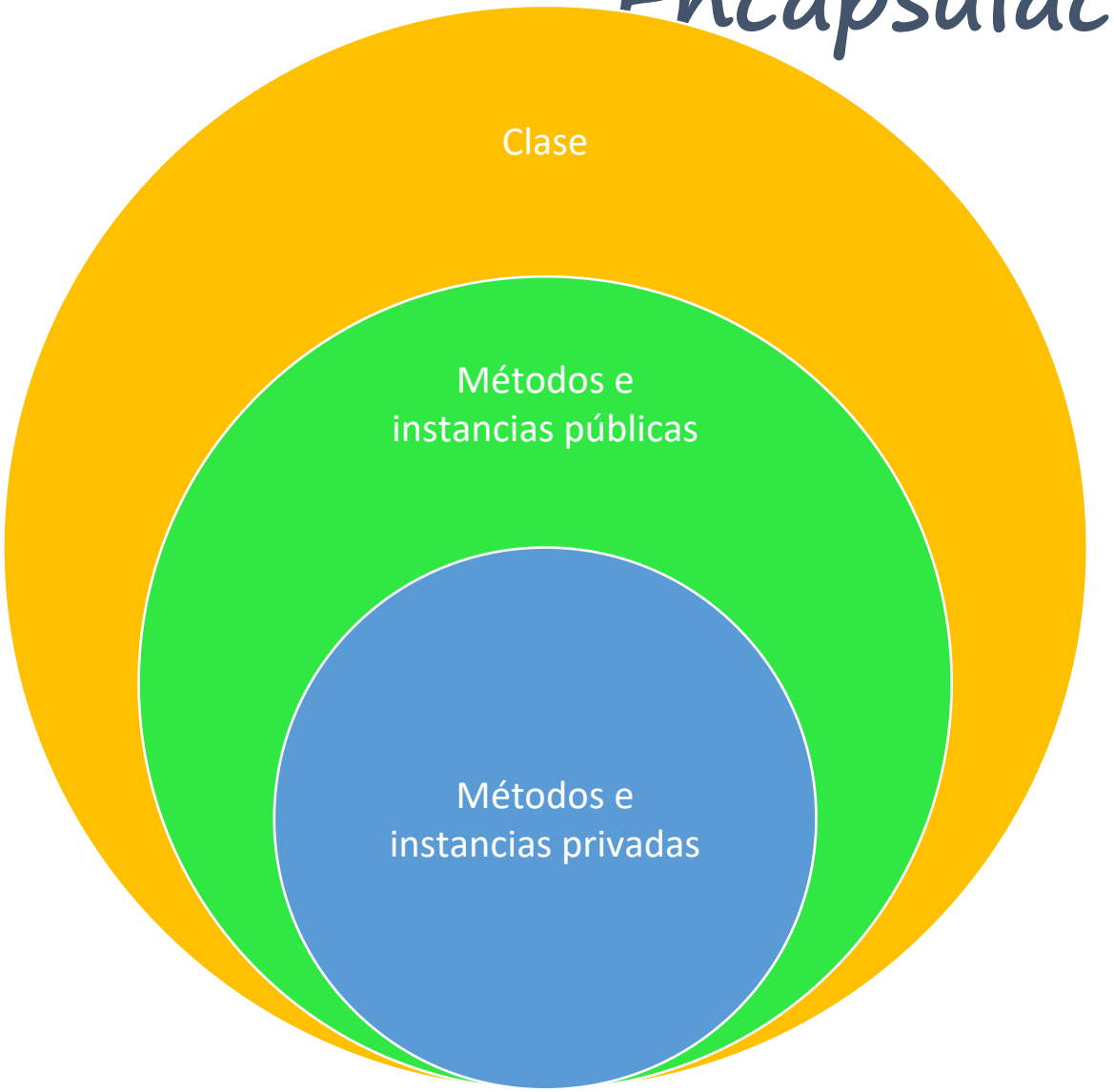
Importante

Identidad Única
(Espacio de Nombres)

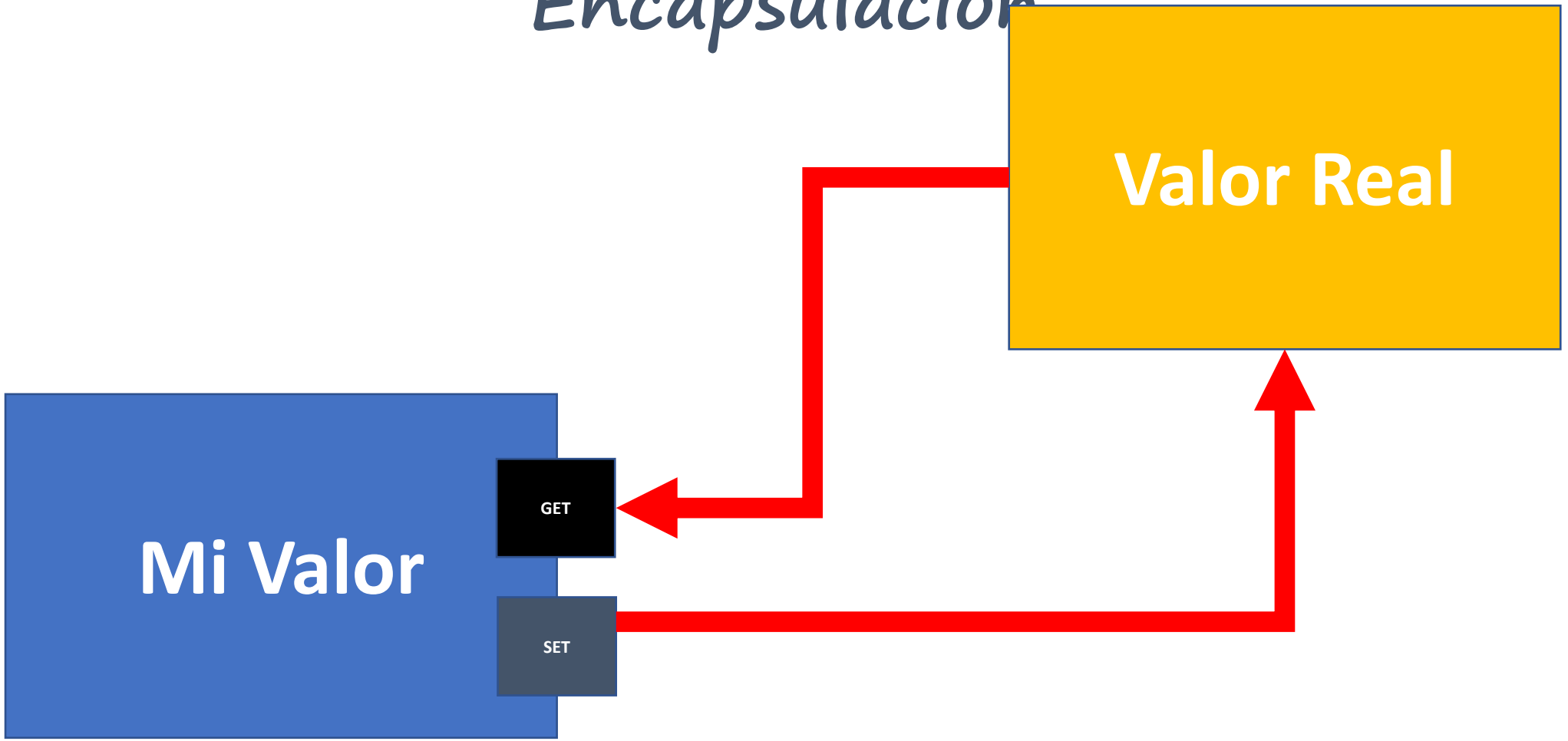
Características del Objeto
(Propiedades/Atributos)

Acciones a Ejecutar
(Métodos/Acciones)

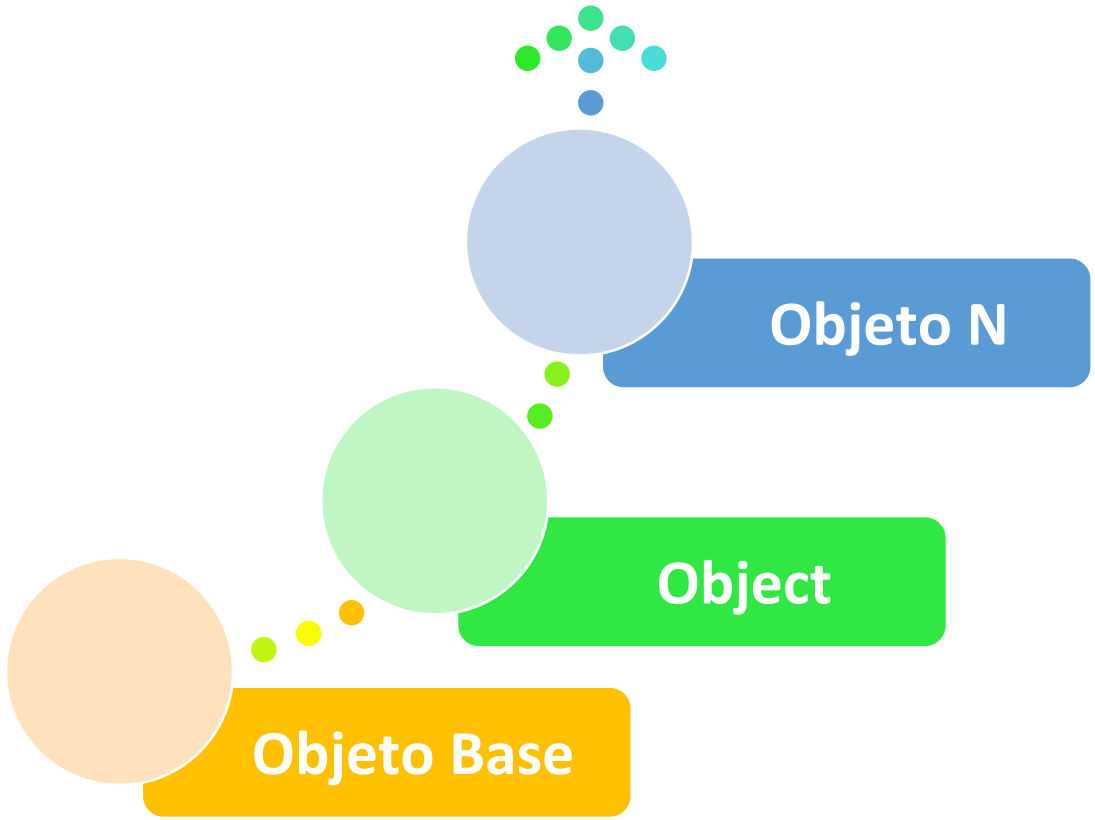
Programación Orientada a Objetos - Encapsulación



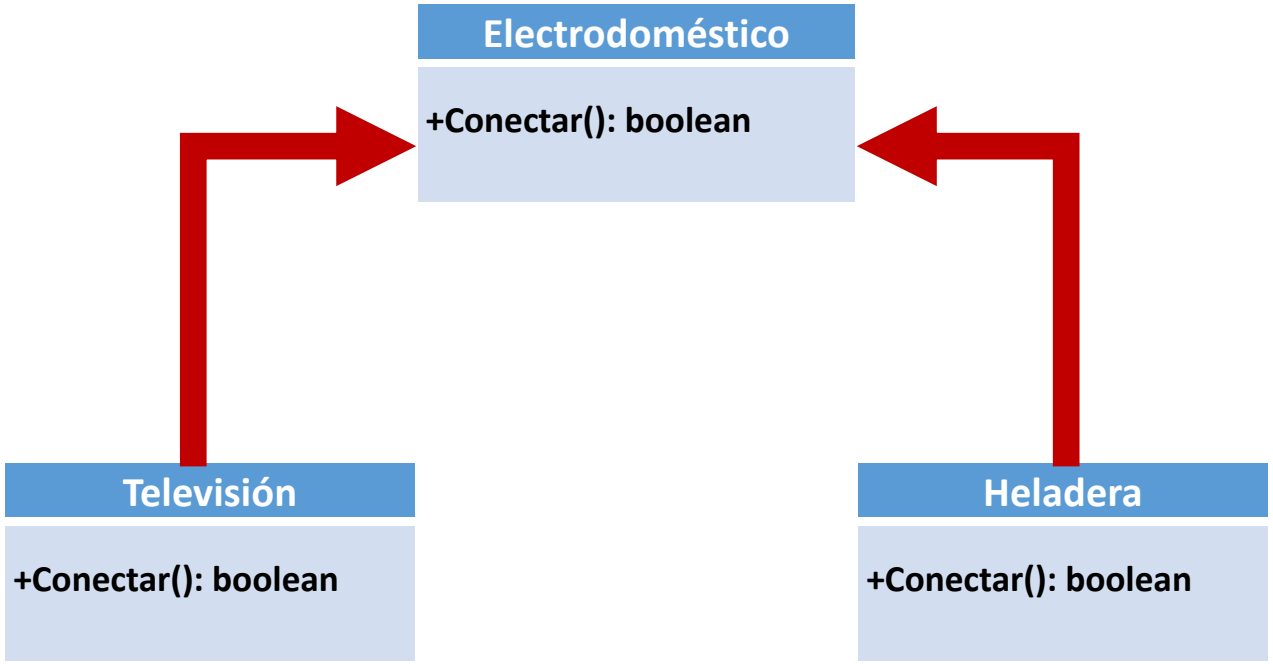
Programación Orientada a Objetos - Encapsulación

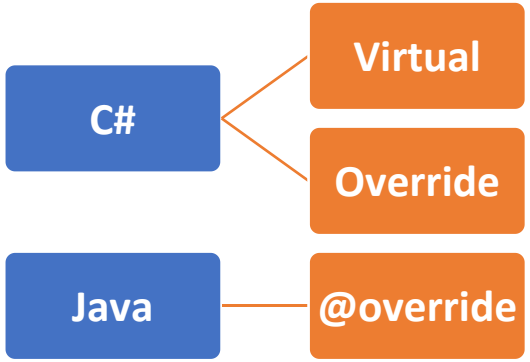


Programación Orientada a Objetos - Herencia

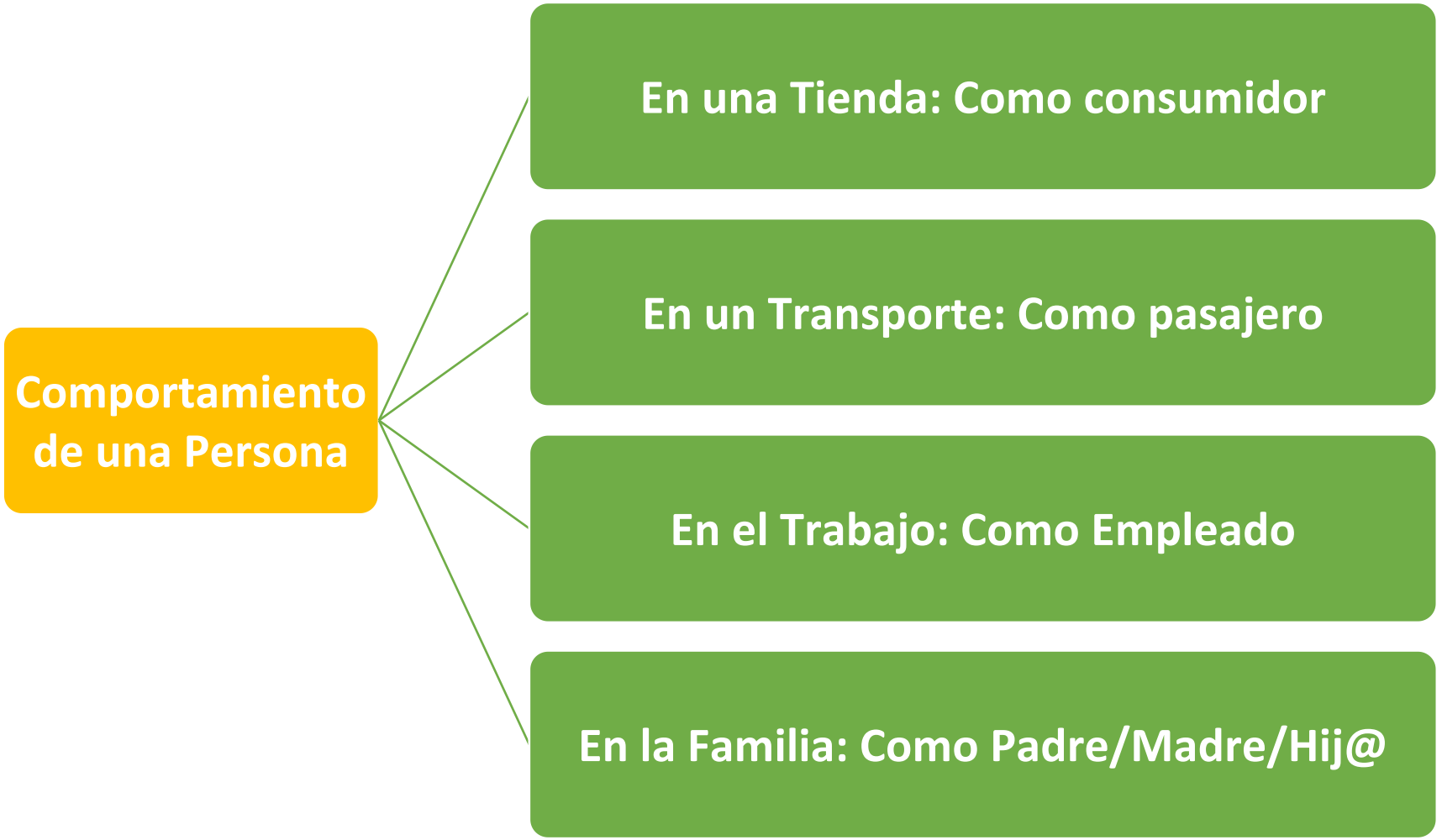


Programación Orientada a Objetos - Polimorfismo

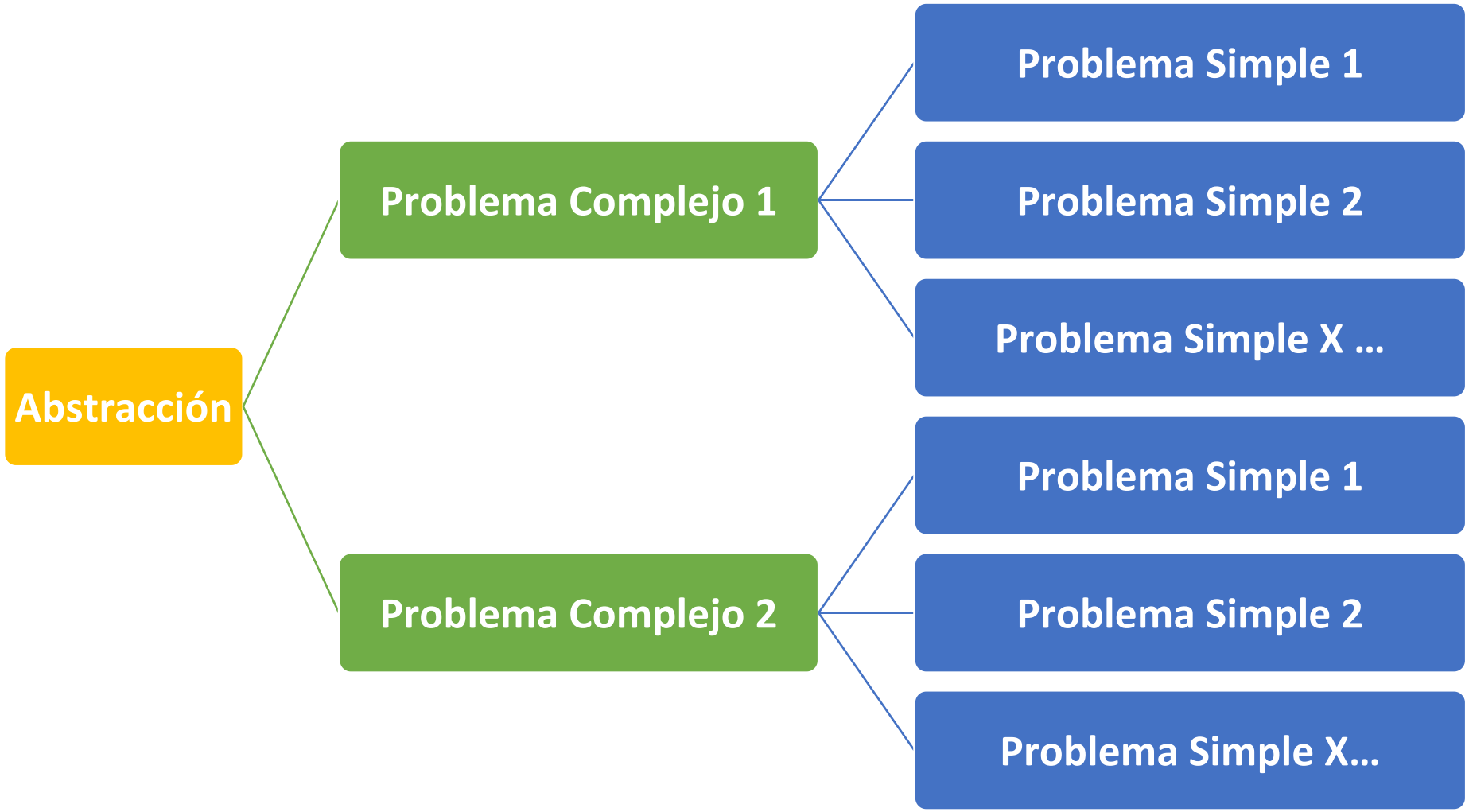




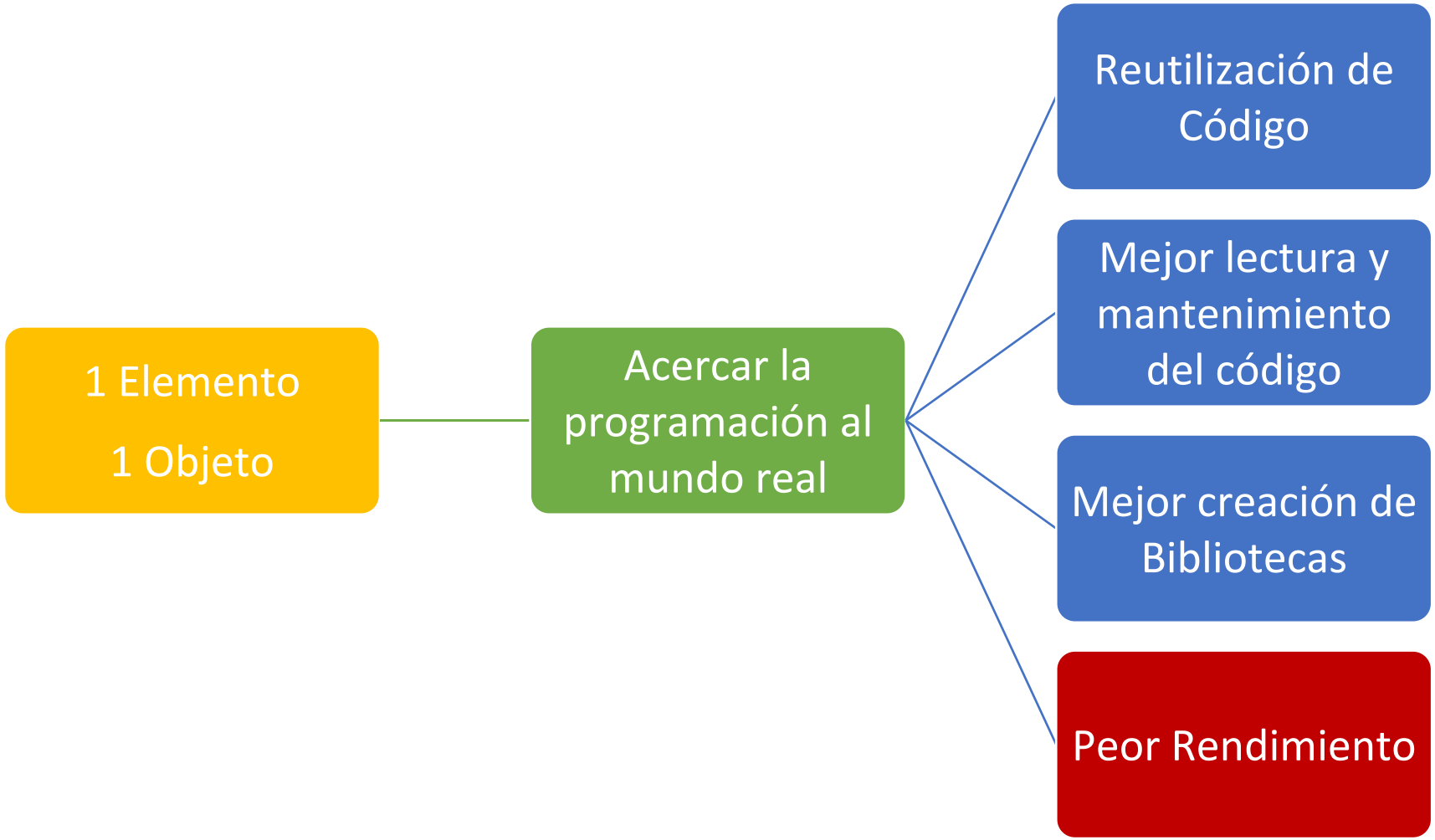
Programación Orientada a Objetos - Polimorfismo



Orientación a Objetos - Abstracción



Principales Ventajas de la POO

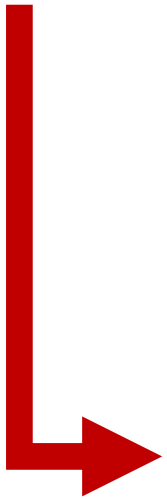


Creación de nuestra primera clase

Producto
+ codigo : int + nombre : string + precio : decimal

```
public class Producto
{
        private int codigo;
        private string nombre;
        private decimal precio;

        public int Codigo { get => codigo; set => codigo = value; }
        public string Nombre { get => nombre; set => nombre = value; }
        public decimal Precio { get => precio; set => precio = value; }
}
```



```
public int Codigo
{
        get { return codigo; }
        set { codigo = value; }
}
```

Ejemplo de Programación Orientada a Objetos

Definición de Clase - Estructura

```
class Persona
        {
         }
```

Ejemplo de Programación Orientada a Objetos

Definición de Atributos - Características

```
public class Producto
{
        private int codigo;
        private string nombre;
        private decimal precio;

        public int Codigo { get => codigo; set => codigo = value; }
        public string Nombre { get => nombre; set => nombre = value; }
        public decimal Precio { get => precio; set => precio = value; }
}
```

Ejemplo de Programación Orientada a Objetos

Definición de Atributos - Métodos

```
class Persona
      {
        string nombre;
        int ojos, brazos, piernas;
        string color_ojos;
        string color_cabellos;

          void andar(int velocidad)
          {
               // código fuente
          }
          void hablar()
          {
               // código fuente
          }
          void comer()
          {
               // código fuente
          }

       }
```

Ejemplo de Programación Orientada a Objetos

Usando la Clase Persona

```
Using System;
Using System.Collection.Generic;
Using System.Linq;
Using System.Text;
Using System.Threading.Tasks;

Namespace POO
{
        class Program
        {
                static void Main(string[] args)
                {
                        Persona p = new Persona();
                        p.hablar();
                }
        }
}
```

Ejemplo de Programación Orientada a Objetos

Definición de Atributos – Métodos públicos

```
class Persona
      {
        string nombre;
        int ojos, brazos, piernas;
        string color_ojos;
        string color_cabellos;

          public void andar(int velocidad)
          {
               // código fuente
          }
          public void hablar()
          {
               // código fuente
          }
          public void comer()
          {
               // código fuente
          }

       }
```

Ejemplo de Programación Orientada a Objetos

Getters y Setters

```
public string getNombre()
        {
            return nombre; // Devuelve el nombre de la Persona
        }

public void setNombre(string nombre)
        {
            this.nombre = nombre; //Modifica, Establece el nombre de la Persona
        }
```

Ejemplo de Programación Orientada a Objetos

Ejemplo - Getters y Setters

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
   namespace POO
  {
      class Program
      {
          static void Main(string[] args)
          {
              Persona p = new Persona();
              p.setNombre(“Juan Manuel Pérez");
              Console.WriteLine(“La Persona: " + p.getNombre() + " es muy agradable");
              Console.ReadKey();
          }
      }
  }
```

Ejemplo de Programación Orientada a Objetos

Getters y Setters - Constructores

```
using System;
  using System.Collections.Generic;
  using System.Linq;
  using System.Text;
  using System.Threading.Tasks;

   namespace POO
  {
      public class Persona
      {
           public Persona(string cabelo) //Valor obrigatório
          {
              Ojos = 2; //Valor default
              Brazos = 2;
              Piernas = 2;
              Color_Cabello = cabelo;
           }
          public Persona() {
            }
          public string Nombre { get; set; }
          public int Ojos { get; set; }
          public string Color_Cabello { get; set; }
          public int Brazos { get; set; }
          public int Piernas { get; set; }
       }
}
```

Ejemplo de Programación Orientada a Objetos

Getters y Setters - Final

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

  namespace POO
  {
      class Program
      {
          static void Main(string[] args)
          {
              Persona p = new Persona();
              p.Nombre = “Juan Manuel Pérez"; //UTILIZANDO CONSTRUTOR DEFAULT
              p.Brazos = 2;
              p.Piernas = 2;
              p.Ojos = 2;
              p.Color_Cabello = “Moreno";
              Console.WriteLine(p.Nombre + " possui " + p.Brazos + " brazos," + "\n" + p.Piernas + " piernas, \n " + p.Ojos + " ojos y cabello " + p.Color_Cabello +"\n");

              Persona p1 = new Persona(“Rubio");  //UTILIZANDO CONSTRUTOR PERSONALIZADO
              p1.Nombre = “Paco";

              Console.WriteLine(p1.Nombre+" tiene "+p1.Brazos+ " brazos"+",\n"+p1.Piernas+" piernas, \n "+p1.Ojos+ " ojos y cabello "+p1.Color_Cabello);


              Console.ReadKey();
          }
      }
  }
```