

09 Herencia

Néstor Suat-Rojas. Ing. Msc (c)
nestor.suat@unillanos.edu.co

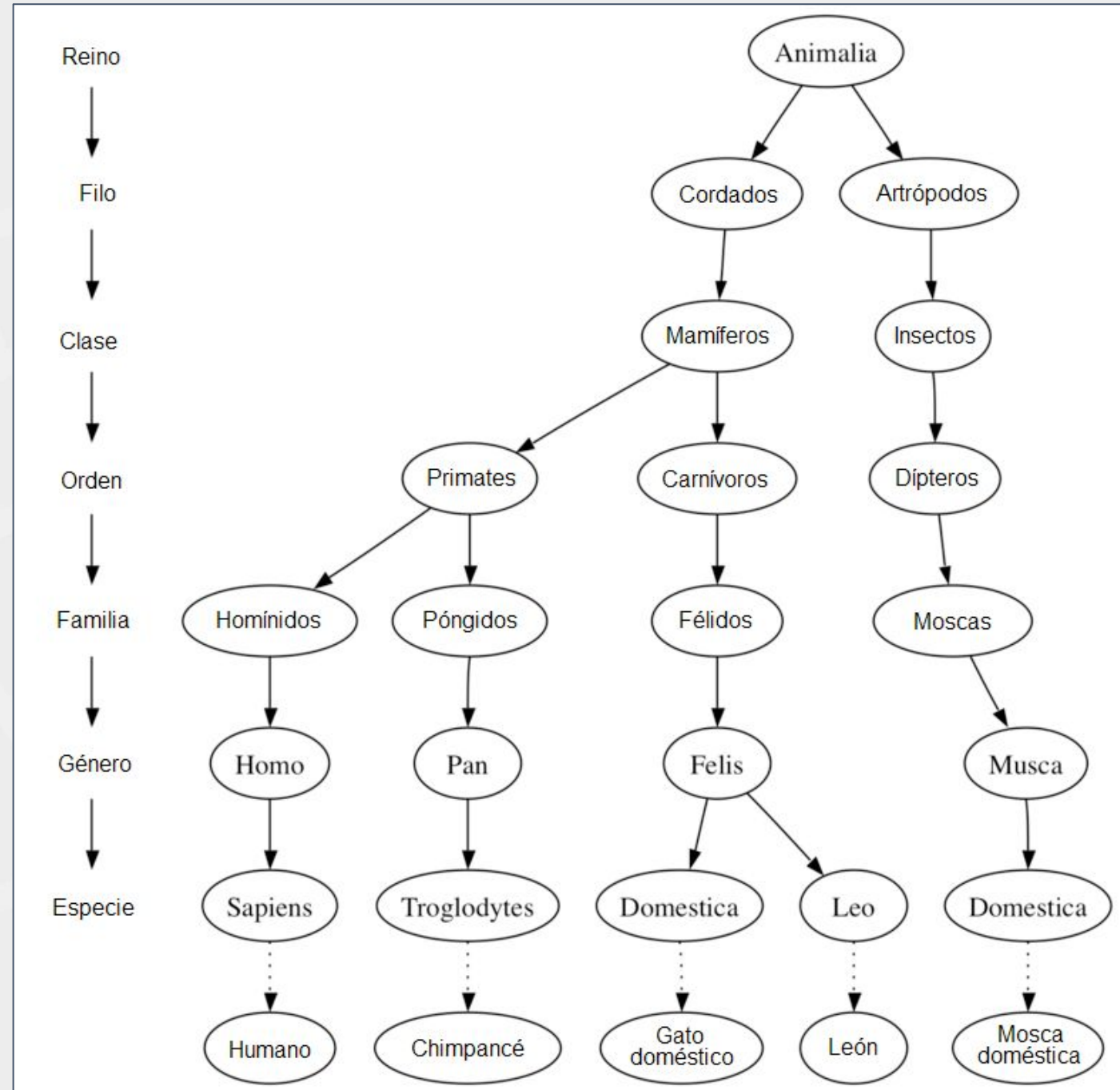
Escuela de Ingeniería
Facultad de Ciencias Básicas e Ingeniería

Introducción

Jerarquización

- Es un proceso por el cual se crean organizaciones de elementos en distinto niveles.
- 1. **Composición:** Cuando unos elementos podemos decir que están compuestos de otros.
- 2. **Clasificación:** Indica que unos elementos son una especialización de otros.

Introducción



Introducción

Reutilización del código

- DRY o Don't Repeat Yourself (No te repitas)
- No es mejor programador quien más líneas de código hace, sino quien mejor las **reutiliza**.
- Evitar escribir dos veces el mismo código, evitar los copia/pega.

Herencia

- ★ Transmisión de código entre unas clases y otras.
- ★ Clase Padre (o Base) y las clases hijas (Derivada).
 - La clase padre es la que transmite su código a las clases hijas.
 - No se necesita volver a incorporar el código de Padre e Hija.

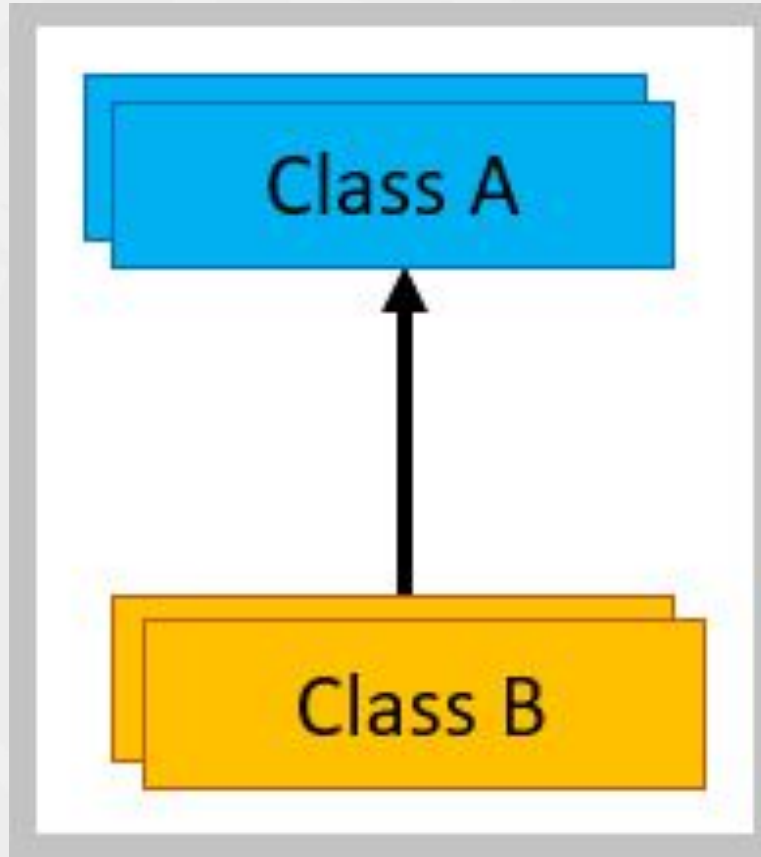
Herencia



El hijo hereda las propiedades de la clase padre

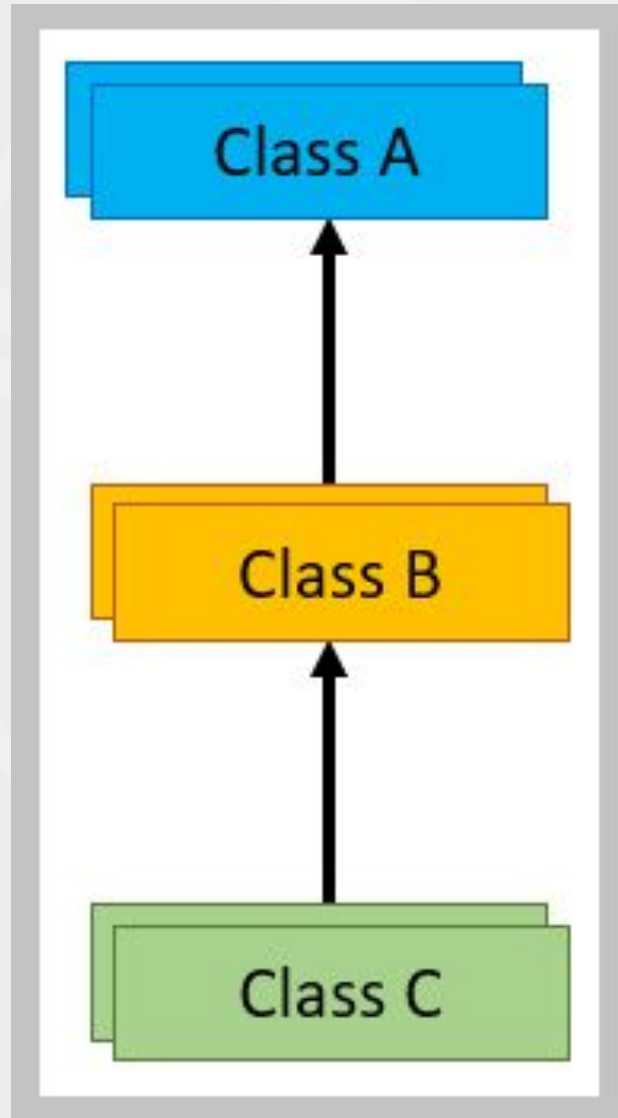
Tipos de Herencia

★ Herencia Simple



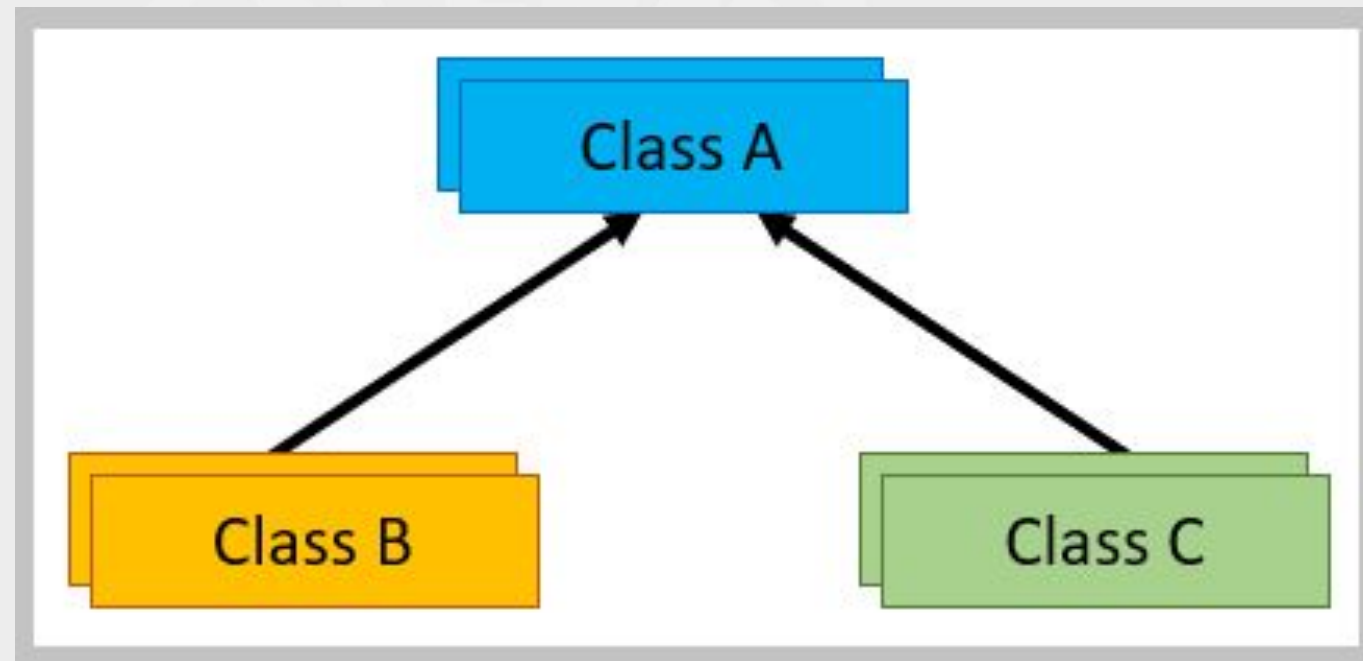
Tipos de Herencia

★ Herencia Multinivel



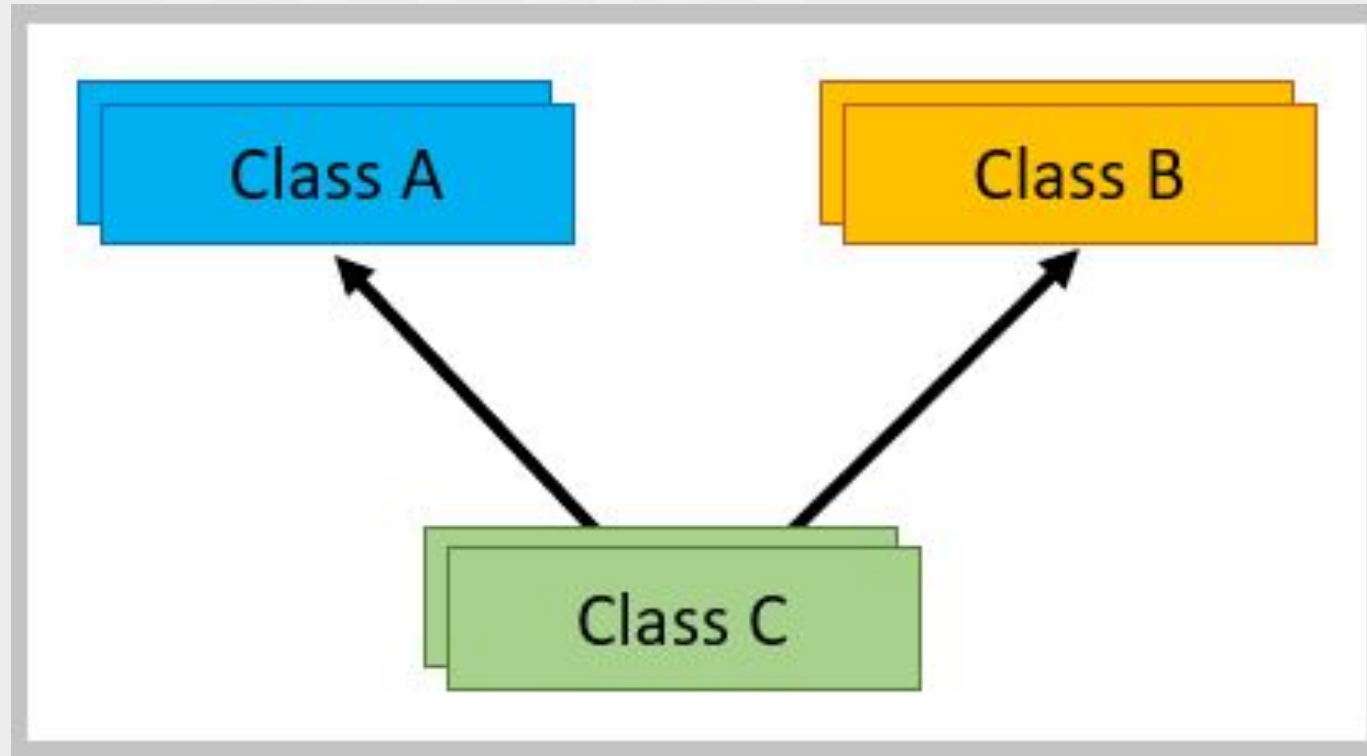
Tipos de Herencia

★ Herencia Jerárquica



Tipos de Herencia

★ Herencia Múltiple



Asociaciones

Otros tipos de asociaciones

- ★ **Agregación**
- ★ **Composición**

Alcance y visibilidad

Tipos de restricción

Tipo de restricciones de acceso

Hay 3 tipos de acceso en C++.

Tipo	Propia clase	Clase derivada	Main()
Public	Si	Si	Si
Protected	Si	Si	No
Private	Si	No	No

Modos de herencia

Imagen tomada de: www.geeksforgeeks.org

CPP

```
// C++ Implementation to show that a derived class
// doesn't inherit access to private data members.
// However, it does inherit a full parent object
class A
{
public:
    int x;
protected:
    int y;
private:
    int z;
};

class B : public A
{
    // x is public
    // y is protected
    // z is not accessible from B
};

class C : protected A
{
    // x is protected
    // y is protected
    // z is not accessible from C
};

class D : private A    // 'private' is default for classes
{
    // x is private
    // y is private
    // z is not accessible from D
};
```

Bibliografía

- Desarrollo web
- Beginners Book
- Software Testing Material



Gracias...