

Universidad de Los Andes
Escuela de Ingeniería de Sistemas
Departamento de Computación

PROGRAMACIÓN 2

Clase 08

Junior Altamiranda
altamira@ula.ve

AGENDA

Programación Orientada a Objetos (POO)

- Herencia.

Ejercicio.

Ejercicio

Realizar un programa en C++ que declare la clase Persona, con campos nombre, apellido, cedula, dirección, telefono; y haga uso de un objeto persona.

Uso = Declarar + Leer + Mostrar.

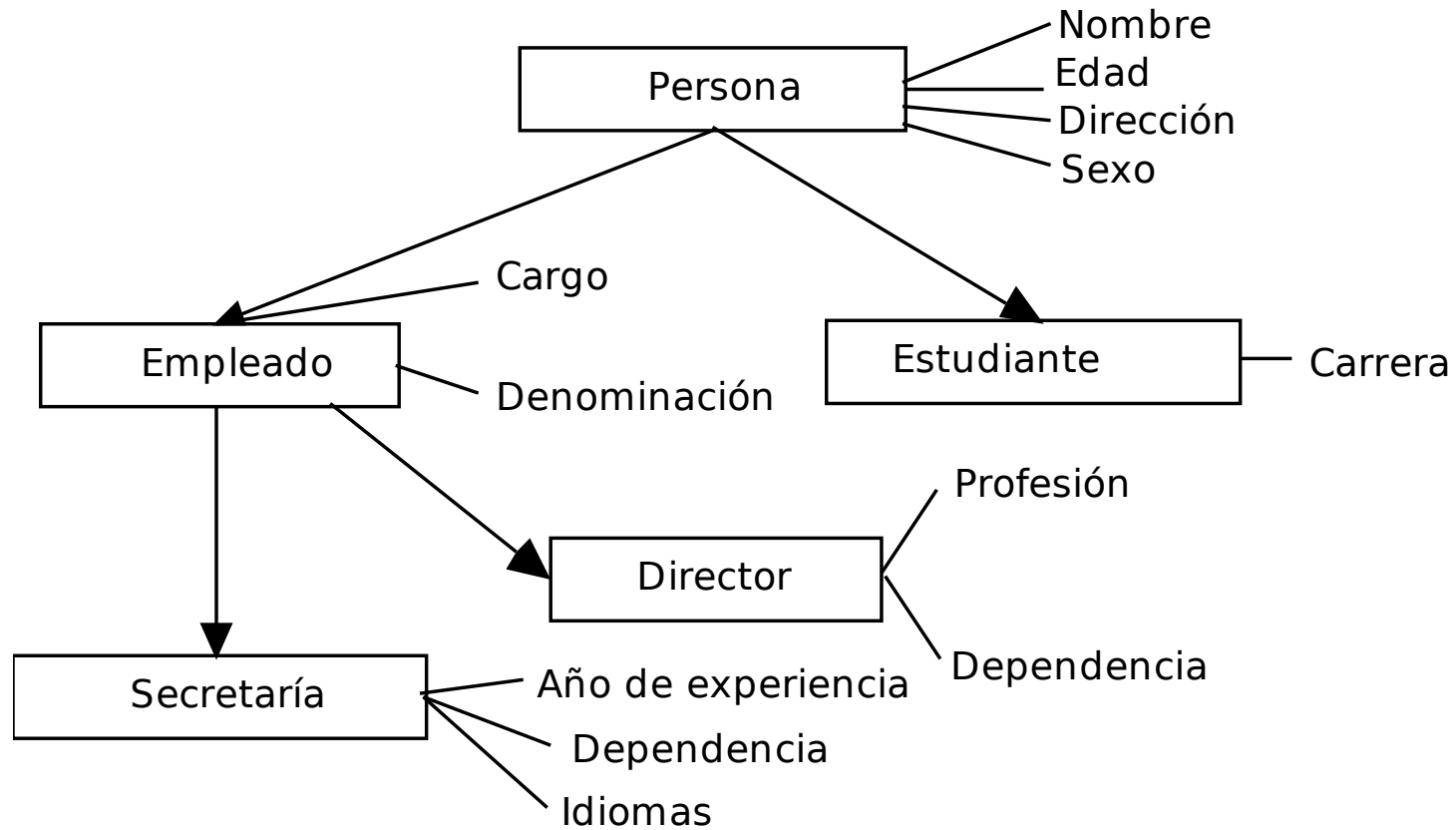
Herencia

- Es una forma de reutilización de software que ofrece la POO
- Permite definir nuevas clases (**Clases derivadas**) a partir de clases ya existentes (**Clases base**)
- Los atributos y métodos son absorbidos (heredados) por las clases derivadas
- Las nuevas clases pueden:
 - Agregar nuevas propiedades y métodos
 - Redefinir métodos

Herencia

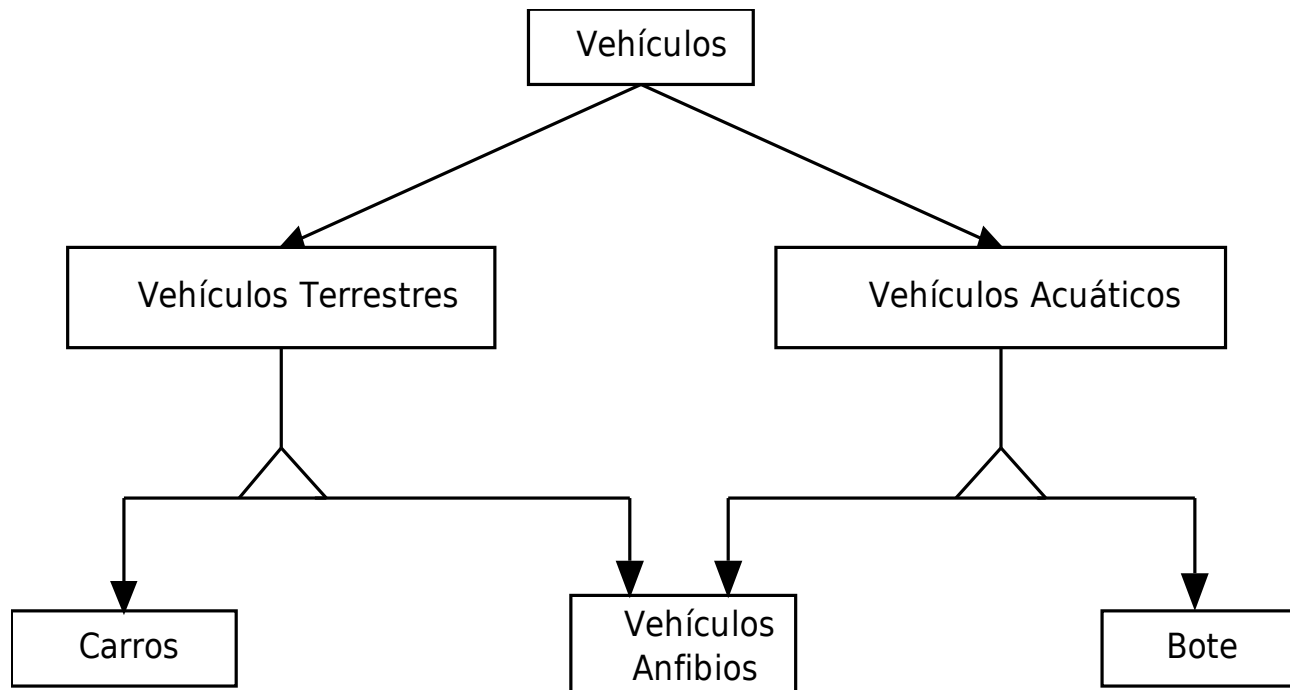
- Una clase derivada es más específica que su clase base
- La clase derivada contiene un menor número de objetos que la clase base
- Una clase derivada no puede acceder a los datos `private` de su clase base. Pero sí a los `public` y `protected`.
- Una clase “cliente” no puede acceder ni a los datos `private` ni a los datos `protected` de un objeto. Sólo puede acceder a los `public`.

Herencia



Tipos de Herencia

- **Herencia simple**
- **Herencia múltiple:** La clase derivada hereda los miembros de varias clases base



Herencia – Implementación en C++

```
class clase_derivada : permiso clase_base {  
    . . .  
};
```

Ejemplo:

```
class apartamento : public inmueble {  
    . . .  
};
```


Herencia – Implementación en C++

- Los constructores y destructores, no se heredan
- Se pueden reutilizar constructores y destructores de la clase base, pero hay que indicarlo explícitamente:

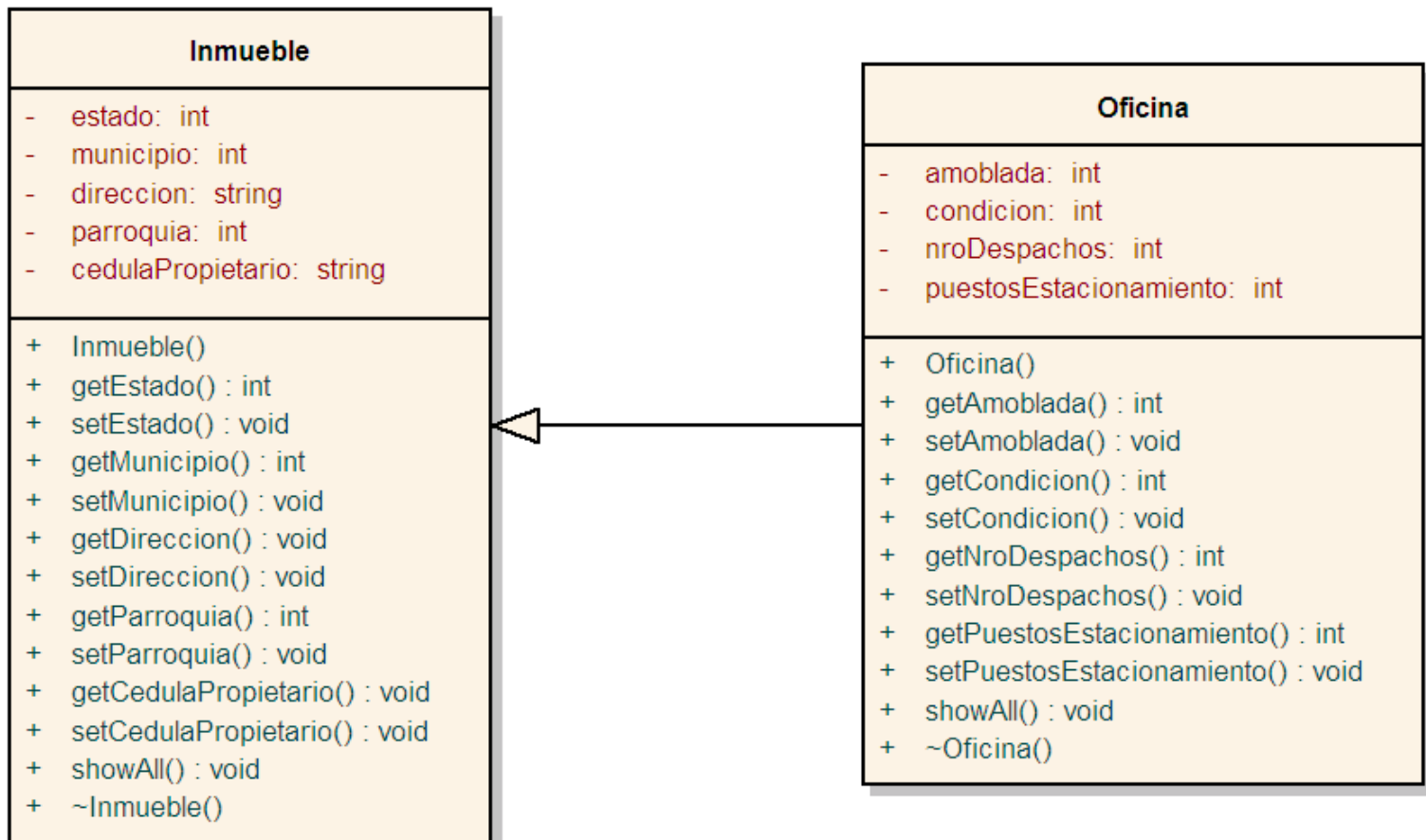
```
clase_base(int a) { }
```

```
clase_derivada(int a, int b) : clase_base(a) { }
```

- Si se invocan constructores o destructores base:
El constructor base es llamado antes que el derivado
El destructor base es llamado después que el derivado

Ejercicio

Realizar la implementación de las clases inmueble y oficina, descritas en el siguiente diagrama de clases:



Ejercicio

Realizar la implementación de dos clases:

Prenda

Propiedades: Genero, Año, Precio, Color

Métodos: Constructor vacío, constructor paramétrico, lectura, desplegar, getGenero, getColor

Camisa

Propiedades: Manga, Cuello, Dibujo Frontal, Dibujo Espalda,

Métodos: Constructor vacío, constructor paramétrico, lectura, desplegar, getManga y getDibujo en la Espalda. Vector dinámico para camisas de mujer, azules, manga corta y con dibujo en la espalda

Camisa es una subclase de Prenda, debe llamar al constructor y destructor de Prenda

Escribir un programa principal que permita registrar cierta cantidad de camisas.