



El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

‘Misión
TIC2022’

HERENCIA



El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

«Misión TIC 2022»

Inducción

www.upb.edu.co/es/mision-tic





El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

‘Misión
TIC 2022’

CICLO II: Programación Básica en Java



El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

‘Misión
TIC2022’

Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Definir y aplicar los conceptos relacionados con herencia en POO
2. Especificar los contextos donde se puede aplicar la herencia
3. Diseñar la herencia entre clases: definición de superclase y subclases
4. Explicar el uso del tipo de acceso protected y su aplicación en herencia
5. Construir programas en Java con la aplicación de los conceptos básicos de herencia



Herencia - Organización jerárquica

¿Qué es?

- Es un mecanismo para la reutilización de software
- Permiten crear una clase que es capaz de absorber los atributos y métodos de una clase padre

¿Por qué?

- Ahorran tiempo durante el desarrollo.
- Crean estructuras jerárquicas especializadas

¿Para qué?

- Extender comportamientos a otros objetos



El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

‘Misión
TIC 2022’

Herencia – CONCEPTO HERENCIA





El futuro digital
es de todos

MinTIC



Misión
TIC 2022

Herencia - Organización jerárquica

- Permite reutilizar código creando nuevas clases a partir de las existentes (construidas y depuradas)
- Compromete una relación de jerarquía (es-un)
- Una nueva clase se generará agregando atributos y/o código a una clase existente
- Una clase (derivada) puede heredar de otra clase (base):
 - Atributos
 - Métodos

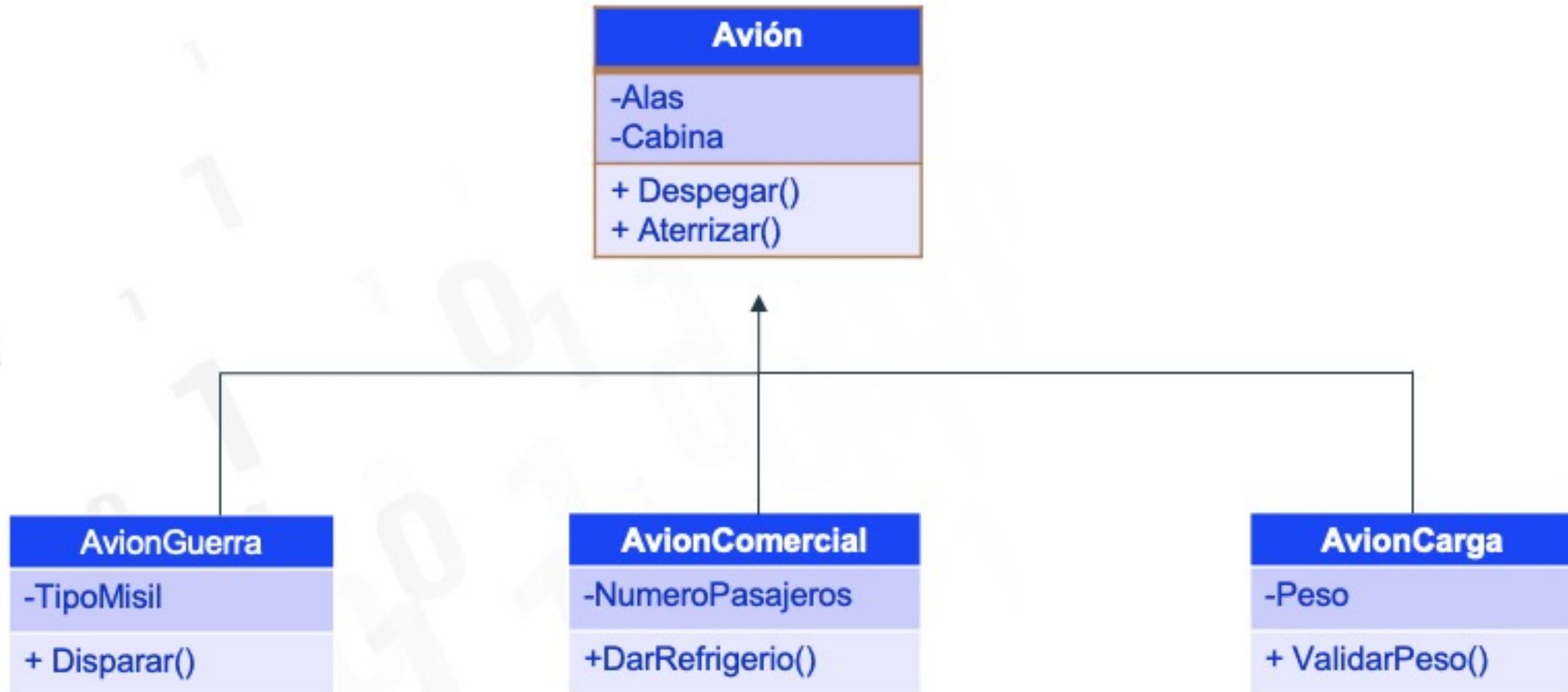


Herencia - Organización jerárquica

- Si una clase **B** hereda de otra clase **A** se tiene que:
 - **B** adopta la estructura (atributos) y comportamiento (métodos) de la clase **A**.
 - **B** puede:
 - agregar nuevos atributos
 - agregar nuevos métodos
 - redefinir métodos



Herencia - Ejemplo:





El futuro digital
es de todos

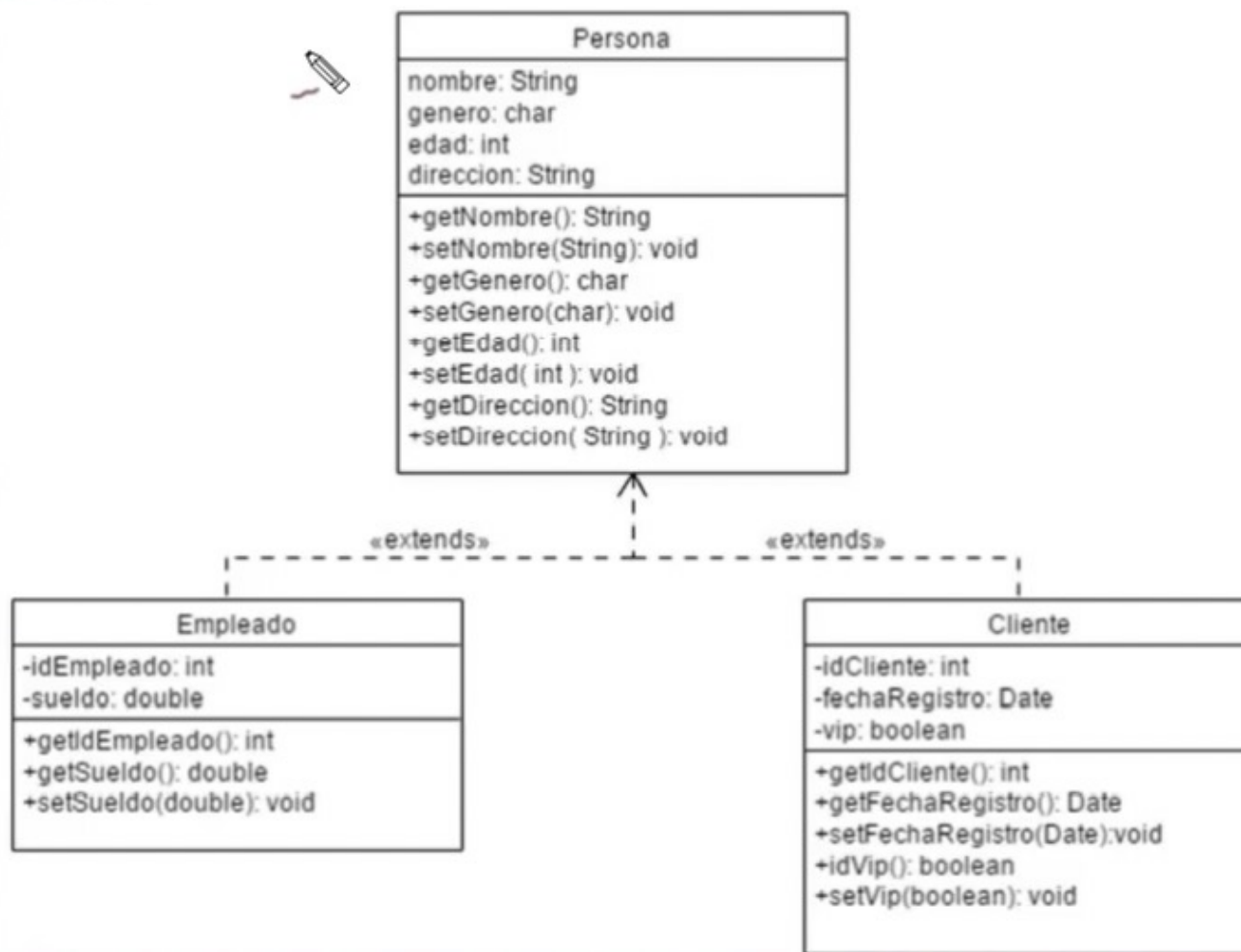
MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

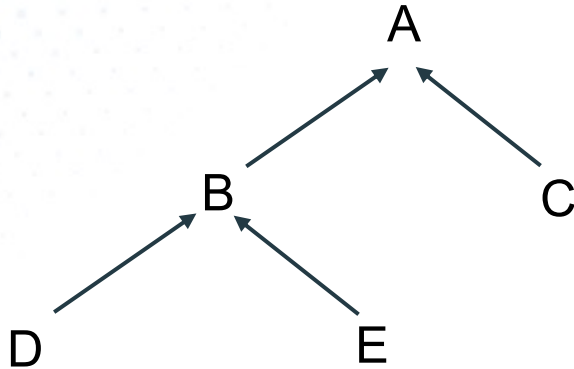
Mision
TIC2022



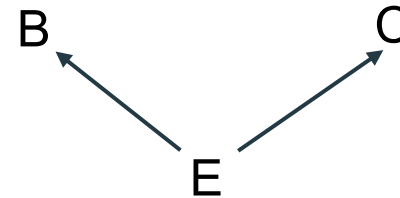


Tipos de Herencia

Herencia Simple: Una clase puede heredar de una única clase.



Herencia Múltiple: Una clase puede heredar de varias clases. Cabe anotar que en **Java no se permite la herencia múltiple.**





El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

‘Misión
TIC2022’

Herencia – Superclases y Subclases

Ejemplo

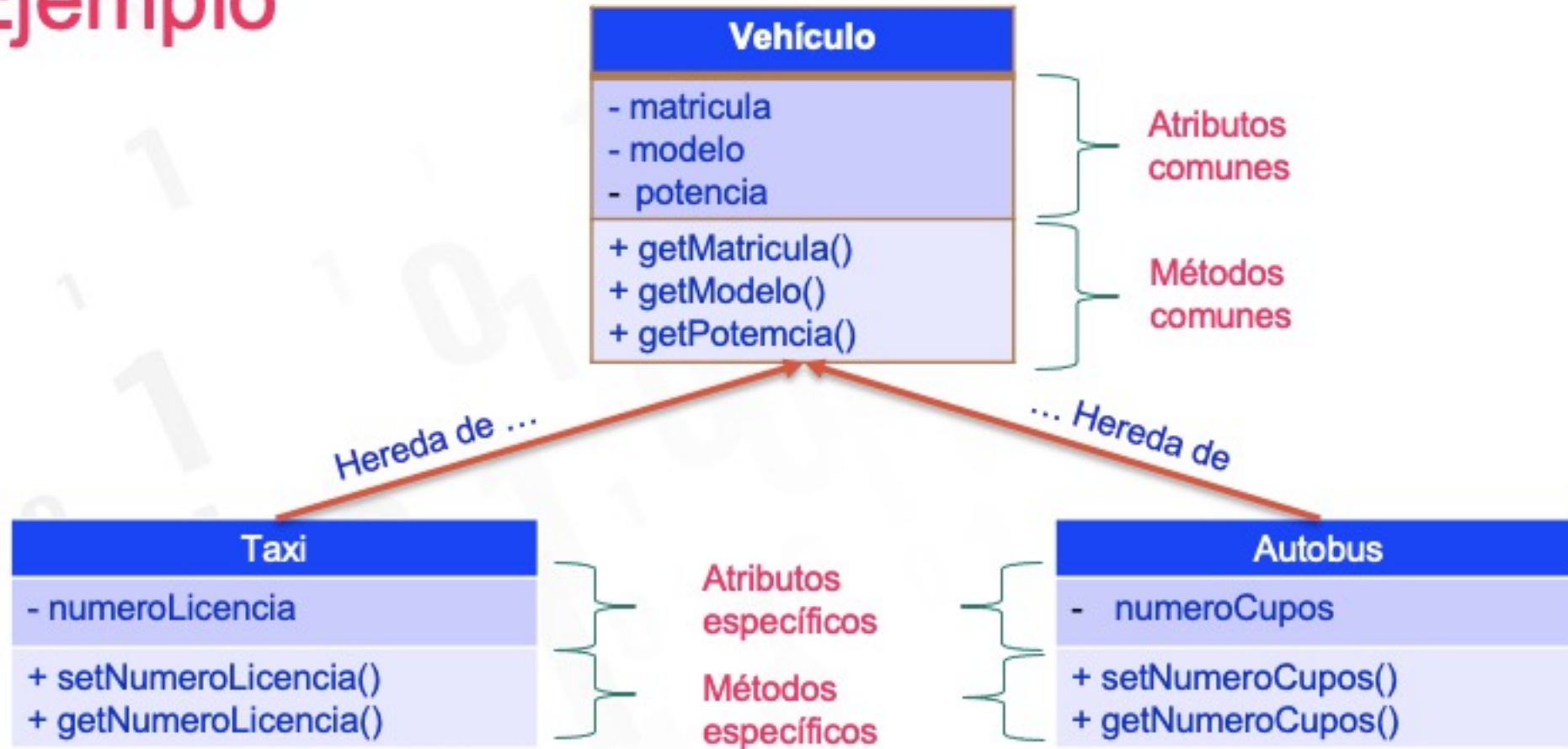
La superclase Vehículo representa a todos los vehículos, como taxis, autobús, barcos, bicicletas, camiones, etcétera.

Por el contrario, la subclase Auto representa a un subconjunto más pequeño y específico de los vehículos.



Herencia – Superclases y Subclases

Ejemplo





Herencia – Superclases y Subclases

Ejemplo de terminología

- Se define la clase Taxi a partir de la clase Vehículo se tiene que:
 - “Taxi” **hereda** las variables y métodos de “Vehiculo”
 - “Taxi” **extiende** de “Vehiculo”
 - “Taxi” es **subclase** de “Vehiculo”
 - clase **derivada**
 - clase **hija**
 - “Vehiculo” es **superclase** de “Taxi”
 - clase **base**
 - clase **padre**
- La herencia realiza la relación **es-un**
 - Un Taxi **es-un** vehículo
 - Un Autobus **es-un** vehiculo



El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

‘Mision
TIC2022’

Herencia – Superclases y Subclases Sintaxis

La sintaxis para declarar subclases es:

```
public class SubClase extends SuperClase{  
  
    ...  
  
}
```



Modificador de acceso protected

- Todos los miembros protected de una superclase conservan su modificador de acceso original cuando se convierten en miembros de la subclase (por ejemplo, los miembros protected de la superclase se vuelven miembros protected de la subclase).
- Los métodos de una subclase pueden referirse a los miembros protected que se hereden de la superclase con sólo utilizar los nombres de los miembros.
- Permite acceder a las propiedades de la superclase o clase padre.
- Fuera de las subclases o clases hijas es equivalente a private.



Modificador de acceso protected

```
public class Ejemplo3  
{  
    protected static int atributo = 0;  
}
```

```
import app.ejemplo3.Ejemplo3;  
public class Ejemplo3_1 extends Ejemplo3  
{  
    public static void main(String[] args)  
    {  
        System.out.println(atributo)  
    }  
}
```



El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

‘Misión
TIC2022’

GRACIAS POR PARTICIPAR