



UTPL
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

Universidad Técnica Particular de Loja
Ingeniería en Computación

Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

Relación entre clases

René Rolando Elizalde Solano
rrelizalde@utpl.edu.ec

Abril - Agosto 2020



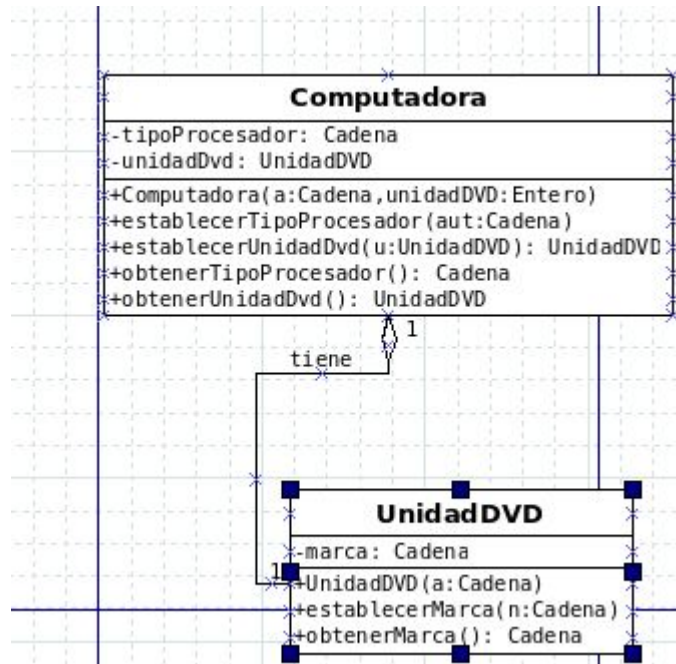
Relación entre clases

- Dos tipo de relaciones agregación y composición.
- Se habla de este tipo de relación, cuando una clase usa otras clases en su estructura.
- Los objetos están compuestos o tiene referencia a otros objetos.
- En los atributos de las clases se usa tipos de dato diferente de los básicos.
- Se puede denotar la relación como tiene-un / tiene-varios.



Agregación

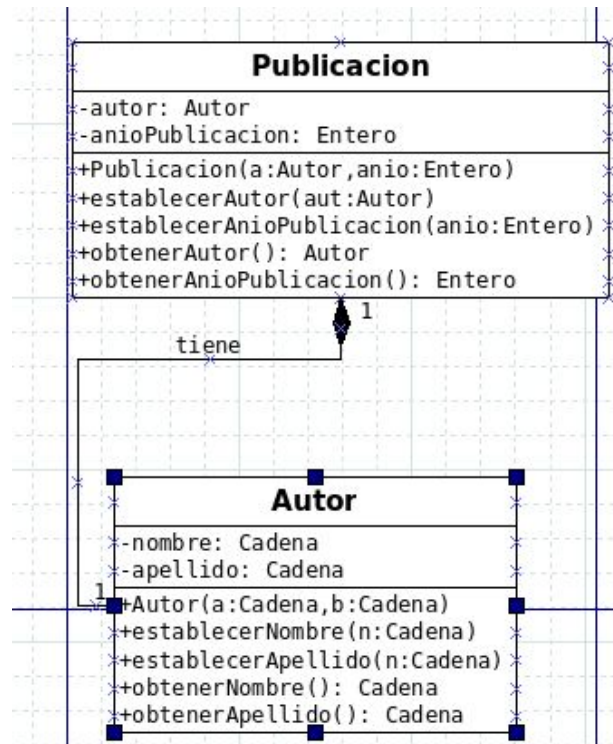
- En UML se representa con una línea que parte desde la clase que actúa como “agregada” y termina en la clase que la contiene. La línea que tiene un diamante en blanco al final (clase que contiene).
- En la agregación la relación es considerada “débil”; ya que la destrucción de la clase que contiene no afecta la clase agregada





Composición

- En UML se representa con una línea que parte desde la clase que actúa como “agregada” y termina en la clase que la contiene. La línea que tiene un diamante relleno al final (clase que contiene).
- En la agregación la relación es considerada “fuerte”; ya que la destrucción de la clase que contiene afecta la clase agregada. Sin la clase contenedora la clase agregada no tendría razón en el contexto.



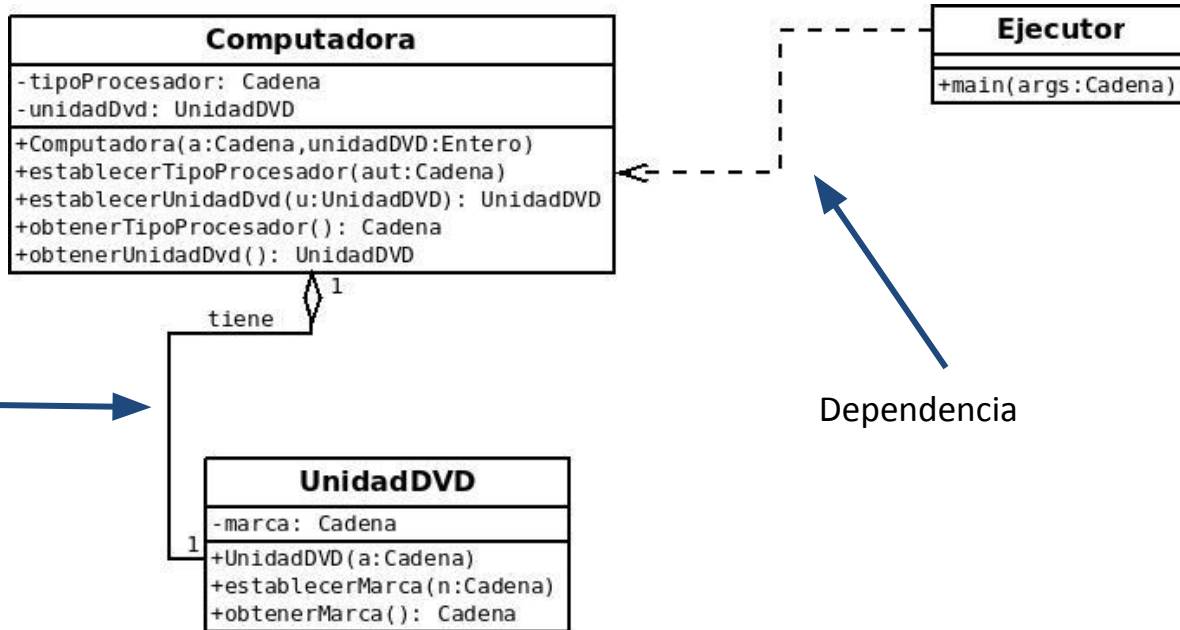


Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

Universidad Técnica Particular de Loja
Ingeniería en Computación

Agregación y Dependencia

Agregación



Dependencia



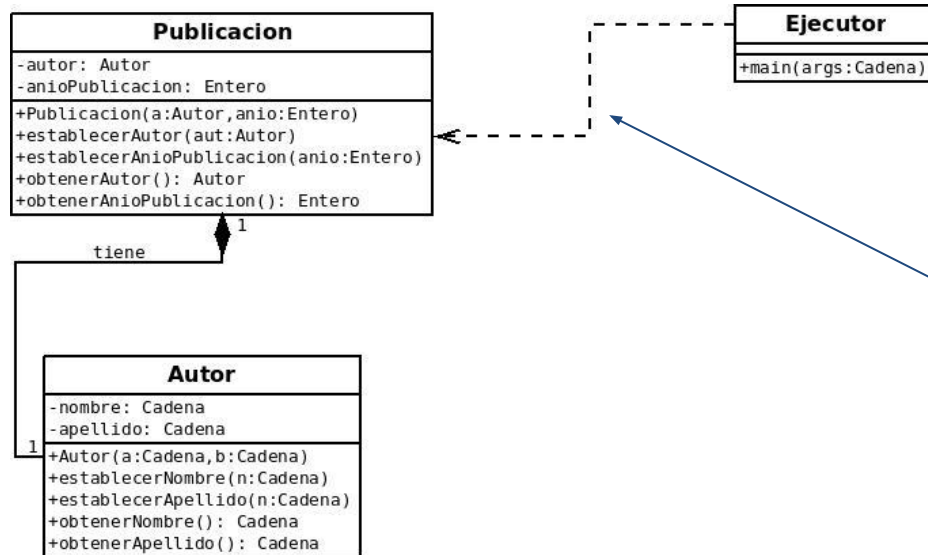
UTPL
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

Universidad Técnica Particular de Loja
Ingeniería en Computación

Composición y Dependencia

Composición



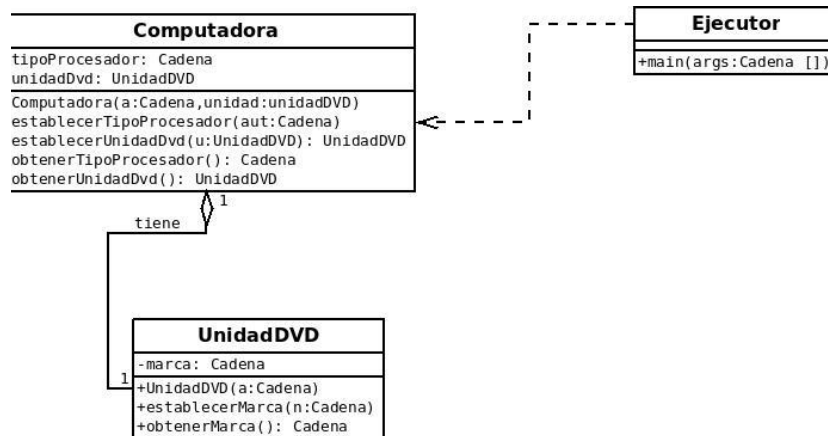
Dependencia



Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

Universidad Técnica Particular de Loja
Ingeniería en Computación

Agregación y Dependencia



```
6 package paqueteuno;
7
8 public class UnidadDVD {
9     private String marca;
10
11     public UnidadDVD(String a){
12         marca = a;
13     }
14
15     public void establecerMarca(String n){
16         marca = n;
17     }
18
19     public String obtenerMarca(){
20         return marca;
21     }
22 }
```

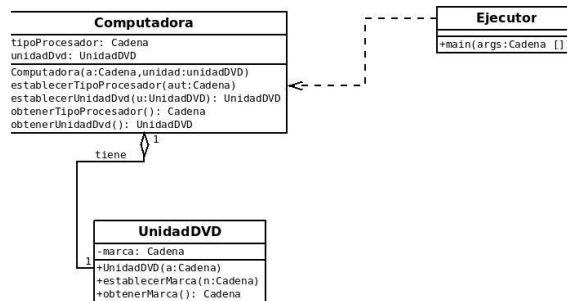
```
12 public class Computadora {
13     private String tipoProcesador;
14     private UnidadDVD unidadDvd;
15
16     public Computadora(String a, UnidadDVD unidad){
17         tipoProcesador = a;
18         unidadDvd = unidad;
19     }
20
21     public void establecerTipoProcesador(String aut){
22         tipoProcesador = aut;
23     }
24
25     public void establecerUnidadDvd(UnidadDVD u){
26         unidadDvd = u;
27     }
28
29     public String obtenerTipoProcesador(){
30         return tipoProcesador;
31     }
32
33     public UnidadDVD obtenerUnidadDvd(){
34         return unidadDvd;
35     }
36 }
```



Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

Agregación y Dependencia

Universidad Técnica Particular de Loja
Ingeniería en Computación



```
12 public class Ejecutor {
13
14     public static void main(String[] args) {
15         // se crea un de tipo UnidadDVD
16         String marca = "Toshiba";
17         UnidadDVD unidad = new UnidadDVD(marca);
18
19         // se crea el objeto de tipo Computadora
20         String procesador = "Intel";
21         Computadora computadora = new Computadora(procesador, unidad);
22         System.out.printf("Computadora:\nProcesador: %s\n"
23             + "Unidad de DVD: %s\n",
24             computadora.obtenerTipoProcesador(),
25             computadora.obtenerUnidadDvd().obtenerMarca());
26     }
27
28 }
```

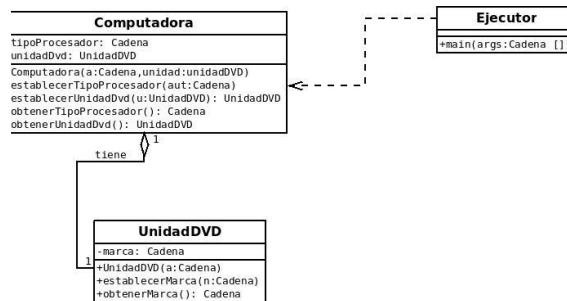
- `computadora.obtenerTipoProcesador();` permite acceder al valor del atributo `tipoProcesador`.
- `computadora.obtenerUnidadDvd().obtenerMarca();` permite acceder al valor de atributo `unidadDvd`, pero `unidadDvd` es otro objeto; y se accede al valor del atributo `marca` a través del método `obtenerMarca`.



Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

Agregación y Dependencia

Universidad Técnica Particular de Loja
Ingeniería en Computación



```
12 public class Ejecutor {
13
14     public static void main(String[] args) {
15         // se crea un de tipo UnidadDVD
16         String marca = "Toshiba";
17         UnidadDVD unidad = new UnidadDVD(marca);
18
19         // se crea el objeto de tipo Computadora
20         String procesador = "Intel";
21         Computadora computadora = new Computadora(procesador, unidad);
22         System.out.printf("Computadora:\nProcesador: %s\n"
23             + "Unidad de DVD: %s\n",
24             computadora.obtenerTipoProcesador(),
25             computadora.obtenerUnidadDvd().obtenerMarca());
26     }
27 }
28
```

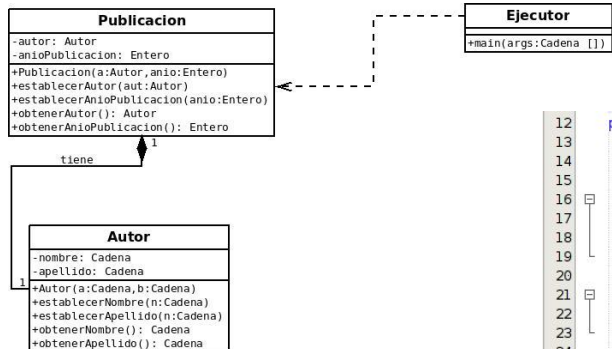
```
run:
Computadora:
Procesador: Intel
Unidad de DVD: Toshiba
```



Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

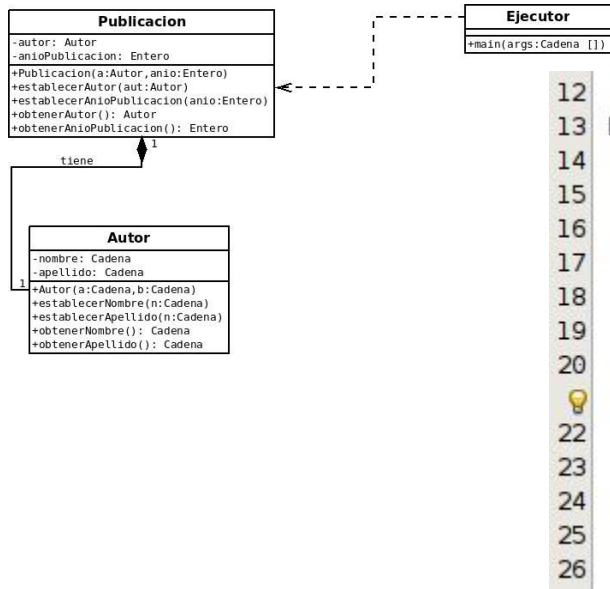
Composición y Dependencia

Universidad Técnica Particular de Loja
Ingeniería en Computación



```
12 public class Autor {
13     private String nombre;
14     private String apellido;
15
16     public Autor(String a, String b){
17         nombre = a;
18         apellido = b;
19     }
20
21     public void establecerNombre(String n){
22         nombre = n;
23     }
24
25     public void establecerApellido(String n){
26         apellido = n;
27     }
28
29     public String obtenerNombre(){
30         return nombre;
31     }
32
33     public String obtenerApellido(){
34         return apellido;
35     }
36 }
```

```
12 public class Publicacion {
13     private Autor autor;
14     private int anioPublicacion;
15
16     public Publicacion(Autor a, int anio){
17         autor = a;
18         anioPublicacion = anio;
19     }
20
21     public void establecerAutor(Autor aut){
22         autor = aut;
23     }
24
25     public void establecerAnioPublicacion(int anio){
26         anioPublicacion = anio;
27     }
28
29     public Autor obtenerAutor(){
30         return autor;
31     }
32
33     public int obtenerAnioPublicacion(){
34         return anioPublicacion;
35     }
36 }
```



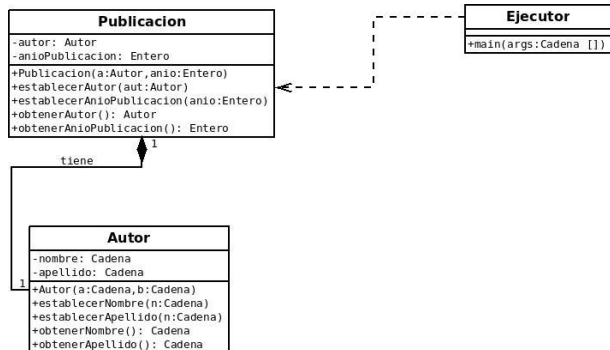
```
12 public class Ejecutor {
13     public static void main(String[] args) {
14         // Se crea un objeto de tipo Autor
15         Autor aut = new Autor("José", "Alvear");
16
17         // Se crea un objeto de tipo Publicación
18         int anioPublicacion = 2000;
19         Publicacion p = new Publicacion(aut, anioPublicacion);
20         System.out.printf("Publicación\nAutor: %s %s\n"
21             + "Año de Publicación: %d\n",
22             p.obtenerAutor().obtenerNombre(),
23             p.obtenerAutor().obtenerApellido(),
24             p.obtenerAnioPublicacion());
25     }
26 }
```



Unidad 2: Estructura y creación de programas en Programación Orientada a Objetos

Composición y Dependencia

Universidad Técnica Particular de Loja
Ingeniería en Computación



```
12 public class Ejecutor {
13     public static void main(String[] args) {
14         // Se crea un objeto de tipo Autor
15         Autor aut = new Autor("José", "Alvear");
16
17         // Se crea un objeto de tipo Publicación
18         int anioPublicacion = 2000;
19         Publicacion p = new Publicacion(aut, anioPublicacion);
20         System.out.printf("Publicación\nAutor: %s %s\n"
21             + "Año de Publicación: %d\n",
22             p.obtenerAutor().obtenerNombre(),
23             p.obtenerAutor().obtenerApellido(),
24             p.obtenerAnioPublicacion());
25     }
26 }
```

```
run:
Publicación
Autor: José Alvear
Año de Publicación: 2000
```



UTPL
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

Universidad Técnica Particular de Loja
Ingeniería en Computación

Gracias