

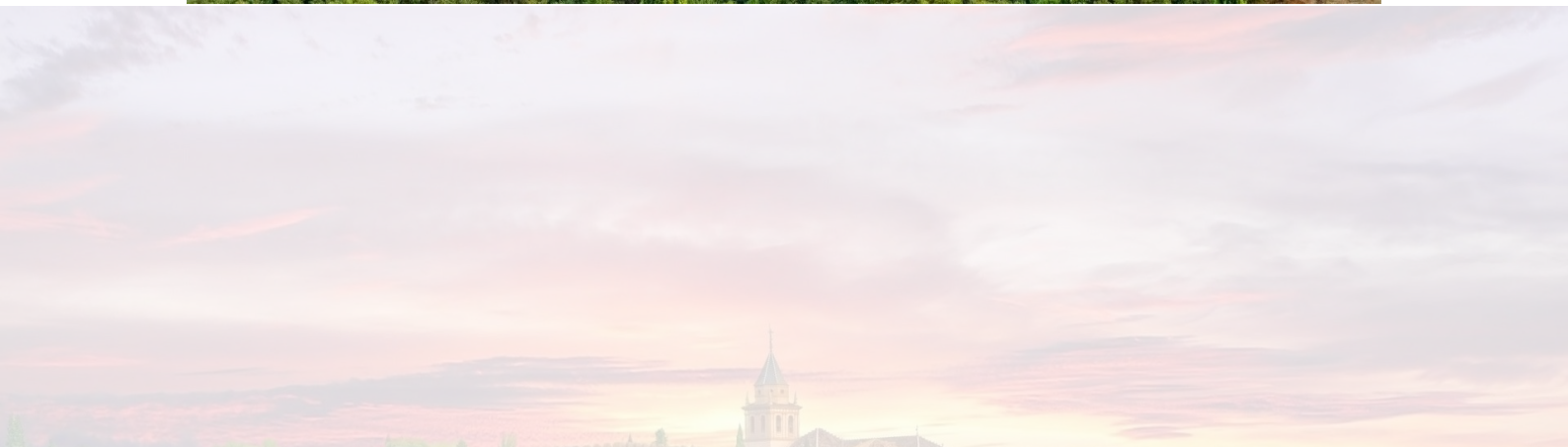


# Ingeniería Informática + ADE

Universidad de Granada (UGR)

**Autor:** Ismael Sallami Moreno

**Asignatura:** Apuntes Visibilidad (PDOO)



# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Java</b>	<b>3</b>
2.1. Private . . . . .	3
2.2. Paquete . . . . .	3
2.3. Protected . . . . .	3
<b>3. Ruby</b>	<b>3</b>
3.1. Private . . . . .	4
3.2. Protected . . . . .	4
<b>4. Ejemplos en Java</b>	<b>4</b>
4.1. Ejemplo 1 . . . . .	4
4.2. Ejemplo 2 . . . . .	5
4.3. Ejemplo 3 . . . . .	5
4.4. Ejemplo 4 . . . . .	6
4.5. Ejemplo 5 . . . . .	7
4.6. Ejemplo 6 . . . . .	7
4.7. Ejemplo 7 . . . . .	8
4.8. Ejemplo 8 . . . . .	9
<b>5. Ejemplos en Ruby</b>	<b>10</b>
5.1. Ejemplo 1 . . . . .	10
5.2. Ejemplo 2 . . . . .	10
5.3. Ejemplo 3 . . . . .	11
5.4. Ejemplo 4 . . . . .	11
5.5. Ejemplo 5 . . . . .	12
5.6. Ejemplo 6 . . . . .	12
5.7. Ejemplo 7 . . . . .	13
5.8. Ejemplo 8 . . . . .	14

## 1 Introducción

- Privado: Solo accesible desde la propia clase.
- Protegido: Solo accesible desde la propia clase y sus subclases.
- Público: Accesible desde cualquier clase.

## 2 Java

### 2.1. Private

- Solo accesible desde la propia clase (Instancia y de clase).
- Se puede acceder a elementos privados de otra instancia distinta si es de la misma clase.

### 2.2. Paquete

- Si no se pone ningún especificador de Visibilidad, por defecto es de paquete.
- son públicos dentro de paquete y privados fuera de él.

### 2.3. Protected

- Públicos dentro del mismo paquete
- Accesible desde subclases de otros paquetes, es decir, dentro de una instancia se puede acceder a los elementos protegidos de sus superclases, con independencia del paquete en el que se encuentren.
- *Si es de una instancia distinta:*
  - Tienes que ser de la misma clase o una subclase.
  - Debe de ser visible por mí.

## 3 Ruby

- Atributos siempre **privados**.
- Métodos siempre **públicos** (se pueden cambiar usando modificadores de acceso).
- Initialize es siempre **privado**.

### 3.1. Private

- Si se hereda se puede acceder a los métodos privados de la superclase (tanto en el ámbito de clase como de instancia).
- No se puede acceder a método privados de clase desde el ámbito de instancia y viceversa.

### 3.2. Protected

- Se puede acceder a los métodos protegidos de la superclase desde la subclase.
- No existen métodos protegidos de clase.

En ruby **una clase y su instancia no son de la misma clase.**

Los atributos @@ **si pueden ser accedidos desde ámbito de instancia.**

## 4 Ejemplos en Java

*Se especifican las soluciones en forma de comentarios.*

### 4.1. Ejemplo 1

```
1 package unPaquete;
2
3 public class Padre {
4     private int privado;           // Campo accesible solo dentro de esta
5                                     // clase.
6     protected int protegido;      // Campo accesible dentro del paquete y
7                                     // subclases.
8     int paquete;                  // Campo accesible solo dentro del
9                                     // paquete.
10    public int publico;            // Campo accesible desde cualquier
11                                     // lugar.
12
13    // Método de instancia que accede a elementos de otra instancia de
14    // la misma clase.
15    public void testInstanciaPadre(Padre o) {
16        System.out.println(o.privado); // Acceso permitido: "privado"
17        // es visible en la misma clase.
18        System.out.println(o.protegido); // Acceso permitido: "protegido"
19        // es visible dentro de la clase.
20        System.out.println(o.paquete); // Acceso permitido: "paquete"
21        // es visible dentro del paquete.
22        System.out.println(o.publico); // Acceso permitido: "publico"
23        // es accesible desde cualquier lugar.
24    }
25
26    // Método estático que intenta acceder a elementos de una instancia
27    .
```



```
18 public static void testClasePadre(Padre o) {
19     System.out.println(o.privado);    // Acceso permitido: "privado"
20     // es visible dentro de la misma clase.
21     System.out.println(o.protegido); // Acceso permitido: "protegido"
22     // es visible dentro de la clase.
23     System.out.println(o.paquete);   // Acceso permitido: "paquete"
24     // es visible dentro del paquete.
25     System.out.println(o.publico);   // Acceso permitido: "publico"
26     // es accesible desde cualquier lugar.
27 }
28 }
```

## 4.2. Ejemplo 2

```
1 package unPaquete;
2
3 public class HijaPaquete extends Padre {
4
5     // Método de instancia que accede a elementos de la superclase y
6     // otra instancia.
7     public void testInstanciaHijaPaquete(Padre o) {
8         System.out.println(privado);    // Error: "privado" no es
9         // accesible fuera de su clase.
10        System.out.println(o.privado);   // Error: "privado" no es
11        // accesible fuera de su clase.
12
13        System.out.println(protegido);   // Acceso permitido: "
14        // protegido" es visible en subclases.
15        System.out.println(o.protegido); // Acceso permitido: "protegido"
16        // es visible en el mismo paquete.
17
18        System.out.println(o.paquete);   // Acceso permitido: "paquete"
19        // es visible dentro del paquete.
20        System.out.println(o.publico);   // Acceso permitido: "publico"
21        // es accesible desde cualquier lugar.
22    }
23
24    // Método estático que accede a elementos de una instancia.
25    public static void testClaseHijaPaquete(Padre o) {
26        System.out.println(o.privado);   // Error: "privado" no es
27        // accesible fuera de su clase.
28        System.out.println(o.protegido); // Acceso permitido: "protegido"
29        // es visible en el mismo paquete.
30        System.out.println(o.paquete);   // Acceso permitido: "paquete"
31        // es visible dentro del paquete.
32        System.out.println(o.publico);   // Acceso permitido: "publico"
33        // es accesible desde cualquier lugar.
34    }
35 }
```

## 4.3. Ejemplo 3

```
1 package otroPaquete;
2
3
4 public class HijaOtroPaquete extends Padre {
5
6     public void testInstanciaHijaOtroPaquete(Padre o) {
7         // Acceso a elementos heredados
8         System.out.println(privado); // (Línea 7) Error: "privado" no es
9         // accesible fuera de la clase.
10        System.out.println(paquete); // (Línea 8) Error: "paquete" no es
11        // accesible desde otro paquete.
12        System.out.println(protogado); // (Línea 9) Acceso permitido: "
13        // protegido" es visible en subclases.
14
15        // Acceso a elementos de otra instancia
16        System.out.println(o.privado); // (Línea 12) Error: "privado" no
17        // es accesible desde otra instancia.
18        System.out.println(o.protogado); // (Línea 13) Error: "protogado" no
19        // es accesible desde otra instancia en otro paquete.
20        System.out.println(o.paquete); // (Línea 14) Error: "paquete" no
21        // es accesible desde otro paquete.
22        System.out.println(o.publico); // (Línea 15) Acceso permitido: "
23        // publico" es accesible desde cualquier lugar.
24    }
25
26    public static void testClaseHijaOtroPaquete(Padre o) {
27        // Acceso a elementos de otra instancia
28        System.out.println(o.privado); // (Línea 20) Error: "privado" no
29        // es accesible desde otra clase.
30        System.out.println(o.protogado); // (Línea 21) Error: "protogado" no
31        // es accesible desde otra instancia en otro paquete.
32        System.out.println(o.paquete); // (Línea 22) Error: "paquete" no
33        // es accesible desde otro paquete.
34        System.out.println(o.publico); // (Línea 23) Acceso permitido: "
35        // publico" es accesible desde cualquier lugar.
36    }
37 }
```

## 4.4. Ejemplo 4

```
1 package otroPaquete;
2
3
4 public class HijaOtroPaquete extends Padre {
5
6     public void testInstanciaHijaOtroPaquete(HijaOtroPaquete o) {
7         System.out.println(o.privado); // (Línea 8) Error: "privado" no
8         // es accesible fuera de la clase.
9         System.out.println(o.protogado); // (Línea 9) Acceso permitido: "
10        // protegido" es visible en subclases.
11        System.out.println(o.paquete); // (Línea 10) Error: "paquete" no
12        // es accesible desde otro paquete.
13    }
```

```

10      System.out.println(o.publico); // (Línea 11) Acceso permitido: "
      publico" es accesible desde cualquier lugar.
11  }
12
13  public static void testClaseHijaOtroPaquete(HijaOtroPaquete o) {
14      System.out.println(o.privado); // (Línea 15) Error: "privado" no
      es accesible desde otra clase.
15      System.out.println(o.protegido); // (Línea 16) Acceso permitido: "
      protegido" es visible en subclases.
16      System.out.println(o.paquete); // (Línea 17) Error: "paquete" no
      es accesible desde otro paquete.
17      System.out.println(o.publico); // (Línea 18) Acceso permitido: "
      publico" es accesible desde cualquier lugar.
18  }
19 }

```

## 4.5. Ejemplo 5

```

1
2 package otroPaquete;
3
4 public class HijaOtroPaquete extends Padre {
5
6     public void testInstanciaHijaOtroPaquete(NietaOtroPaquete o) {
7         System.out.println(o.privado); // (Línea 8) Error: "privado" no
            es accesible fuera de la clase.
8         System.out.println(o.protegido); // (Línea 9) Error: "protegido" no
            es accesible en instancias externas.
9         System.out.println(o.paquete); // (Línea 10) Error: "paquete" no
            es accesible desde otro paquete.
10        System.out.println(o.publico); // (Línea 11) Acceso permitido: "
            publico" es accesible desde cualquier lugar.
11    }
12
13    public static void testClaseHijaOtroPaquete(NietaOtroPaquete o) {
14        System.out.println(o.privado); // (Línea 15) Error: "privado" no
            es accesible desde otra clase.
15        System.out.println(o.protegido); // (Línea 16) Error: "protegido" no
            es accesible en instancias externas.
16        System.out.println(o.paquete); // (Línea 17) Error: "paquete" no
            es accesible desde otro paquete.
17        System.out.println(o.publico); // (Línea 18) Acceso permitido: "
            publico" es accesible desde cualquier lugar.
18    }
19 }

```

## 4.6. Ejemplo 6

```

1 package otroPaquete;
2
3 public class NietaOtroPaquete extends HijaOtroPaquete {
4

```

```
5 // Ahora probamos con un parámetro de la superclase
6 public void testInstanciaNietaOtroPaquete(HijaOtroPaquete o) {
7     // No se puede acceder a 'privado' de otra instancia
8     // System.out.println(o.privado);
9     // No se puede acceder a 'protegido' de otra instancia en un
    paquete diferente
10    // System.out.println(o.protegido);
11    // No se puede acceder a 'paquete' de otra instancia porque no
    está en el mismo paquete
12    // System.out.println(o.paquete);
13    System.out.println(o.publico); // Se puede acceder porque es pú
    blico
14 }
15
16 public static void testClaseNietaOtroPaquete(HijaOtroPaquete o) {
17     // No se puede acceder a 'privado' de otra instancia
18     // System.out.println(o.privado);
19     // No se puede acceder a 'protegido' de otra instancia en un
    paquete diferente
20    // System.out.println(o.protegido);
21    // No se puede acceder a 'paquete' de otra instancia porque no
    está en el mismo paquete
22    // System.out.println(o.paquete);
23    System.out.println(o.publico); // Se puede acceder porque es pú
    blico
24 }
25 }
```

## 4.7. Ejemplo 7

```
1 package otroPaquete;
2
3 public class NietaOtroPaquete extends HijaOtroPaquete {
4
5     public void testInstanciaNietaOtroPaquete(NietaOtroPaquete o) {
6         // No se puede acceder a 'privado' de otra instancia
7         // System.out.println(o.privado);
8         System.out.println(o.protegido); // Se puede acceder porque es
    protegido
9         // No se puede acceder a 'paquete' de otra instancia porque no
    está en el mismo paquete
10        // System.out.println(o.paquete);
11        System.out.println(o.publico); // Se puede acceder porque es pú
    blico
12    }
13
14    public static void testClaseNietaOtroPaquete(NietaOtroPaquete o) {
15        // No se puede acceder a 'privado' de otra instancia
16        // System.out.println(o.privado);
17        System.out.println(o.protegido); // Se puede acceder porque es
    protegido
18        // No se puede acceder a 'paquete' de otra instancia porque no
    está en el mismo paquete
19        // System.out.println(o.paquete);
```



```
20         System.out.println(o.publico); // Se puede acceder porque es pú  
21             blico  
22     }
```

## 4.8. Ejemplo 8

```
1 // En el paquete base  
2 package base;  
3  
4 public class A {  
5     protected int protegidoA = 0;  
6 }  
7  
8 // En el paquete base  
9 public class B extends A {  
10     protected int protegidoB = 1;  
11 }  
12  
13 // En el paquete base2  
14 package base2;  
15 import base.*;  
16  
17 public class C extends B {  
18     protected int protegidoC = 2;  
19  
20     public void test() {  
21         A a = new A();  
22         // No se puede acceder a 'protegidoA' de una instancia de A en  
23         // un paquete diferente  
24         // a.protegidoA = 666;  
25  
26         B b = new B();  
27         // No se puede acceder a 'protegidoB' de una instancia de B en  
28         // un paquete diferente  
29         // b.protegidoB = 666;  
30  
31         C c = new C();  
32         c.protegidoA = 555; // Se puede acceder porque C hereda de A  
33         c.protegidoB = 555; // Se puede acceder porque C hereda de B  
34  
35         D d = new D();  
36         d.protegidoA = 555; // Se puede acceder porque D hereda de A  
37         d.protegidoB = 555; // Se puede acceder porque D hereda de B  
38         d.protegidoD = 555; // Se puede acceder porque es miembro de D  
39  
40         D d2 = new D();  
41         d2.protegidoB = 555;  
42         // No se puede acceder a 'protegidoD' de otra instancia en un  
43         // paquete diferente  
44         // d2.protegidoD = 555;  
45  
46         this.protegidoA = 777; // Se puede acceder porque C hereda de A  
47         this.protegidoB = 777; // Se puede acceder porque C hereda de B
```

## 5 EJEMPLOS EN RUBY

```
45         this.protegidoC = 777; // Se puede acceder porque es miembro de
46             C
47     }
48 }
49 // En el paquete base2
50 package base2;
51
52 public class D extends C {
53     protected int protegidoD = 3;
54 }
```

## 5 Ejemplos en Ruby

### 5.1. Ejemplo 1

```
1  class Padre
2      private
3      def privado
4          end
5
6      protected
7      def protegido
8          end
9
10     public
11     def publico
12         end
13
14     def test(p)
15         privado
16         self.privado # Correcto solo a partir de Ruby 2.7
17         # p.privado # No se puede acceder a un método privado de otra
18                     # instancia
19
20         protegido
21         self.protegido
22         p.protegido # Se puede acceder a un método protegido de otra
23                     # instancia
24     end
25 end
```

### 5.2. Ejemplo 2

```
1  # Fuera de cualquier clase
2
3  Padre.new.test(Padre.new)
4  p = Padre.new
5
6  # Acceso a métodos fuera de la clase o subclases
7
```

```
8 # p.privado # No se puede acceder a un método privado fuera de la clase
9 # p.protegido # No se puede acceder a un método protegido fuera de la
10 # clase o subclases
p.publico # Se puede acceder a un método público
```

### 5.3. Ejemplo 3

```
1 class Hija < Padre
2   def test(p)
3     privado
4     self.privado # Correcto solo a partir de Ruby 2.7
5     # p.privado # No se puede acceder a un método privado de otra
6     # instancia
7
8     protegido
9     self.protegido
10    p.protegido # Se puede acceder a un método protegido de otra
11    # instancia
12
13    publico
14    self.publico
15    p.publico
16  end
17 end
18
19 # Fuera de cualquier clase
20 Hija.new.test(Hija.new)
21 Hija.new.test(Padre.new)
22 h = Hija.new
23 # h.privado # No se puede acceder a un método privado fuera de la clase
24 # h.protegido # No se puede acceder a un método protegido fuera de la
25 # clase o subclases
26 h.publico # Se puede acceder a un método público
```

### 5.4. Ejemplo 4

```
1 class Padre
2   private
3   def privado_instancia
4     end
5
6   def self.privado_clase
7     end
8
9   private_class_method :privado_clase
10
11   public
12   def test
13     # No se puede acceder a un método privado de clase con un receptor
14     # explícito
15     # self.class.privado_clase
16   end
17 end
```

```
15   end
16
17   def self.test(p)
18     # No se puede acceder a un método privado de instancia de otra
19     # instancia
20     # p.privado_instancia
21   end
22 end
23
24 # Fuera de cualquier clase
25 Padre.new.test
26 Padre.test(Padre.new)
```

## 5.5. Ejemplo 5

```
1  class Padre
2    @instanciaDeClase = 1
3    @duda = 2
4    @@deClase = 11
5    @@duda = 22
6
7    def initialize
8      @deInstancia = 333
9      @duda = 444
10   end
11
12   def self.salida
13     # puts @instanciaDeClase + 1
14     # No se puede acceder a la variable de instancia de la clase desde
15     # una subclase
16     # puts @duda + 1 unless @duda.nil? # desde Hija?
17     puts @@deClase + 1
18     puts @@duda + 1
19   end
20
21   def salida
22     #puts @deInstancia + 1
23     #puts @duda + 1
24     # No se puede acceder porque en Ruby todos los atributos son
25     # privados
26     puts @@deClase + 1
27     puts @@duda + 1
28   end
29 end
```

## 5.6. Ejemplo 6

```
1  class Hija < Padre
2    @instanciaDeClase = -1
3
4    # Sobreescribimos el valor fijado anteriormente
```

```
5      # Este atributo es compartido
6      @@deClase = -11
7
8      def modifica
9          # Acceso a los atributos definidos en Padre
10         # No se puede acceder a la variable de instancia de la clase desde
            una instancia
11         # @duda += 111
12     end #deberia de ser self.modifica
13 end
14
15 # Fuera de cualquier clase
16
17 Padre.salida
18 Hija.salida # Atención a lo que ocurre con la segunda línea
19
20 p = Padre.new
21 p.salida
22 h = Hija.new
23 h.salida
24
25 # h.modifica debería de ser self.modifica
26 h.salida
```

## 5.7. Ejemplo 7

```
1  class A
2      protected
3      def protegidoA
4          end
5  end
6
7  class B < A
8      protected
9      def protegidoB
10         end
11  end
12
13  class C < B
14      protected
15      def protegidoC
16          end
17
18      public
19      def test
20          A.new.protegidoA
21          B.new.protegidoA
22          B.new.protegidoB
23          C.new.protegidoA
24          C.new.protegidoB
25          C.new.protegidoC
26          D.new.protegidoA
27          # D.new.protegidoD
```



```
28      # No puedo debido a que estoy en C y es D la que hereda de C, puedo
      acceder de D a C, pero no de C a D
29
30  end
31
32  class D < C
33    protected
34    def protegidoD
35    end
36  end
37
38  C.new.test
39  # C.new.protegidoC
40  # No se puede acceder a un método protegido fuera de la clase o
    subclases
```

## 5.8. Ejemplo 8

```
1  class FalsaSeguridad
2    # Un consultor puede ser muy peligroso
3    attr_reader :lista
4
5    def initialize
6      @lista = [1, 2, 3, 4]
7    end
8
9    def info
10     puts @lista.size
11   end
12 end
13
14 # Fuera de cualquier clase
15 f = FalsaSeguridad.new
16 f.info # 4
17
18 # Modificamos el estado interno
19 # Simplemente usando un consultor
20 # Aunque el atributo sea privado
21 # Cuidado con las referencias
22 f.lista.clear
23
24 f.info # 0
```