

# Actividad en Clase: Constructores, sobrecarga de métodos, paquetes y modificadores de acceso

## Parte 1: Verdadero o Falso (justifica tu respuesta brevemente) 5

1. Dos métodos pueden tener el mismo nombre y el mismo número de parámetros si los tipos de estos parámetros son diferentes. Verdadero  
Los métodos sobrecargados cambian tipo y cantidad en los parámetros
2. Un método privado (`private`) puede ser accedido desde cualquier clase del mismo paquete. Falso

Si es privado solamente desde la misma clase.

3. Cuando se sobrecarga un método, se puede cambiar su nombre. Falso  
La sobrecarga debe mantener el nombre

4. Un método sin modificador se puede acceder cualquier clase del mismo paquete. Verdadero

Sin modificador el acceso es a nivel de paquete

5. Todos los métodos definidos en un paquete deben ser públicos (`public`). Falso

Pueden tener el modificador de acuerdo al acceso que se considere

## Parte 2: Análisis de código. Lee el siguiente fragmento de código y responde las preguntas. 3

<pre>package util;  public class Calculadora {      public int sumar(int a, int b) {         return a + b;     }      public double sumar(double a, double b) {         return a + b;     } }</pre>	<ol style="list-style-type: none"><li>6. ¿Cuál es el nombre del paquete donde se encuentra la clase `Calculadora`? util</li><li>7. ¿Cuántas veces se sobrecarga el método `sumar`? Hay una nueva versión del método, está dos veces</li></ol>
---	---

<pre> }  private int restar(int a, int b) {     return a - b; } } </pre>	<p>8. ¿Se puede acceder al método `restar` desde otra clase dentro del mismo paquete? Justifica.</p> <p>No, porque está privado y solo se puede acceder dentro de la misma clase</p>
--	--

### Parte 3: Seleccione los enunciados verdaderos respecto a por qué el constructor es un método especial en Java 6

- a) Porque puede ser llamado explícitamente como cualquier otro método usando su nombre.
- b) Porque lleva el mismo nombre que la clase.
- c) Porque no tiene tipo de retorno.
- d) Porque siempre se declara como public.
- e) Porque es el primer método que se ejecuta al crear una instancia.
- f) Porque el compilador de java puede crear un constructor por defecto.

### Parte 4: Completar código (paquetes y modificadores de acceso)

Completa el siguiente código para que funcione correctamente.

<pre> // Archivo: Mensaje.java // Ubicado en el paquete: herramientas  package herramientas;  public class Mensaje {     public void mostrar() {         System.out.println("¡Hola desde el paquete herramientas!");     } } </pre>	<pre> // Archivo: Main.java // Ubicado en el paquete: app  package app;  import herramientas.Mensaje;  public class Main {     public static void main(String[] args) {         Mensaje m = new Mensaje();         m.mostrar();     } } </pre>
---	--

	}
--	---

1. Escribe las palabras clave faltantes en ambas clases para que funcionen correctamente. 2
2. ¿Qué modificador de acceso usaste para el método `mostrar()` y por qué? 1
3. Si las clases estuvieran en el mismo paquete app que podrías cambiar en el código para mantener la funcionalidad. 3

<pre>// Archivo: Mensaje.java // Ubicado en el paquete: herramientas  package app;  public class Mensaje {     void mostrar() {         System.out.println("¡Hola desde el paquete app!");     } }</pre>	<pre>// Archivo: Main.java // Ubicado en el paquete: app  package app;  public class Main {     public static void main(String[] args) {         Mensaje m = new Mensaje();         m.mostrar();     } }</pre>
--	--

Quitar el import

Quitar el public

Cambiar el nombre del paquete