

App

```
App.java ✖
1 package com.empresa;
2
3 public class App {
4
5     public static void main(String[] args) {
6
7         Canasta canasta = Canasta.getSingletonInstance();
8         canasta.setNombre("jupari");
9
10        ClaseA a = new ClaseA();
11        ClaseB b = new ClaseB();
12
13        a.obtenerCanasta();
14        b.obtenerCanasta();
15
16    }
17
18 }
19
```

Consola

```
<terminated> App [Java Application] C:\Program Files\Java\jre1.8.0_152\bin\javaw.exe
A: jupari
B: jupari
```

Canasta.java - ClaseA.java - ClaseB.java

```
Canasta.java ✖
1 package com.empresa;
2
3 public class Canasta {
4
5     private String nombre;
6     private static Canasta canasta;
7
8     public Canasta() {
9         super();
10    }
11
12    public static Canasta getInstance() {
13        if (canasta == null) {
14            canasta = new Canasta();
15        }
16        return canasta;
17    }
18
19    public String getNombre() {
20        return nombre;
21    }
22
23    public void setNombre(String nombre) {
24        this.nombre = nombre;
25    }
26
27    public static Canasta getCanasta() {
28        return canasta;
29    }
30
31    public static void setCanasta(Canasta canasta) {
32        Canasta.canasta = canasta;
33    }
34
35 }
36
```

ClaseA.java

```
1 package com.empresa;
2
3 public class ClaseA {
4
5     public void obtenerCanasta() {
6         Canasta canasta = Canasta.getSingletonInstance();
7         System.out.println("A: " + canasta.getNombre());
8     }
9
10 }
11
```

ClaseB.java

```
1 package com.empresa;
2
3 public class ClaseB {
4
5     public void obtenerCanasta() {
6         Canasta canasta = Canasta.getSingletonInstance();
7         System.out.println("B: " + canasta.getNombre());
8     }
9
10 }
11
```

Julio Pari (IT Architect IBM)



Si te ha interesado este artículo y deseas un apoyo o asesoría en algún requerimiento, envíame un mensaje a: (info@juliopari.com) o sino a través de LinkedIn: <https://www.linkedin.com/in/juliopari/>
