

# Patrón Singleton en Java, con ejemplos

El proyecto de este post los puedes descargar pulsando [AQUI](https://github.com/jarroba/Singleton_Java) ([https://github.com/jarroba/Singleton\\_Java](https://github.com/jarroba/Singleton_Java)).

En este tutorial vamos a hablar del patrón de diseño "**Singleton**", que en ingeniería del software es un patrón diseñado para limitar la creación de objetos pertenecientes a una clase. El objetivo de este patrón es el de garantizar que una clase solo tenga una instancia (o ejemplar) y proporcionar un punto de acceso global a ella. Esta patrón; por ejemplo, suele ser utilizado para las conexiones a bases de datos.

Este patrón se implementa haciendo privado el constructor de la clase y creando (en la propia clase) un método que crea una instancia del objeto si este no existe. Veamos a continuación un ejemplo de este patrón en el que vamos a crear una clase llamada "**SoyUnico**" con un constructor que va a ser privado y un método llamado '*getSingletonInstance()*' que será el encargado de crear una instancia de esta clase si no se ha creado todavía:

```
public class SoyUnico {

    private String nombre;
    private static SoyUnico soyUnico;

    // El constructor es privado, no permite que se genere un constructor por defecto.
    private SoyUnico(String nombre) {
        this.nombre = nombre;
        System.out.println("Mi nombre es: " + this.nombre);
    }

    public static SoyUnico getSingletonInstance(String nombre) {
        if (soyUnico == null){
            soyUnico = new SoyUnico(nombre);
        }
        else{
            System.out.println("No se puede crear el objeto "+ nombre + " porque ya existe un
objeto de la clase SoyUnico");
        }

        return soyUnico;
    }

    // metodos getter y setter

}
```

Vemos como al llamar al método '*getSingletonInstance()*' pregunta si el atributo '*soyUnico*' es null o no, y si no lo es lo crea. En el caso de que ya haya sido creado no imprimirá un mensaje por pantalla diciendonos que ya existe un objeto de esa clase.

ACEPTAR

Vamos a ver como efectivamente nos crea un solo objeto, ejecutando el siguiente código en el que llamamos dos veces el método '*getSingletonInstance()*':

```
public class Main {  
  
    public static void main(String[] args) {  
  
        SoyUnico ricardo = SoyUnico.getSingletonInstance("Ricardo Moya");  
        SoyUnico ramon = SoyUnico.getSingletonInstance("Ramón Invarato");  
  
        // ricardo y ramon son referencias a un único objeto de la clase SoyUnico  
        System.out.println(ramon.getNombre());  
        System.out.println(ricardo.getNombre());  
    }  
}
```

Como salida a este código tenemos lo siguiente:

```
Mi nombre es: Ricardo Moya  
No se puede crear el objeto Ramón Invarato porque ya existe un objeto de la clase SoyUnico  
Ricardo Moya  
Ricardo Moya
```

Vemos como efectivamente cuando no hay un objeto de la clase "**SoyUnico**", lo crea y si lo quiere volver a crear, el método nos dice que ya hay otro creado pero le asigna la referencia al objeto, es decir que las variable 'ricardo' y 'ramon' referencian al mismo y único objeto de la clase "**SoyUnico**"; tal y como podemos ver al preguntar por el atributo '*nombre*' de la clase.

Por último este patrón no quedaría perfectamente implementado si no controlamos la copia (o clone) de los objetos de esta clase; por ello es importante sobrescribir el método '*clone()*', para que cuando se quiera clonar este objeto se lance una excepción (<https://jarroba.com/excepciones-exception-en-java-con-ejemplos/>). Este método lo sobrescribimos de la siguiente forma:

```
//Sobreescribimos el método clone, para que no se pueda clonar un objeto de esta clase  
@Override  
public SoyUnico clone(){  
    try {  
        throw new CloneNotSupportedException();  
    } catch (CloneNotSupportedException ex) {  
        System.out.println("No se puede clonar un objeto de la clase SoyUnico");  
    }  
    return null;  
}
```

Veamos que pasa si intentamos clonar el objeto de la clase "**SoyUnico**":

#### Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

```
public class Main {

    public static void main(String[] args) {

        SoyUnico ricardo = SoyUnico.getInstance("Ricardo Moya");

        try{
            SoyUnico richard = ricardo.clone();
            System.out.println(richard.getNombre());
        }catch (NullPointerException ex){
            ex.printStackTrace();
        }
    }
}
```

Vemos que nos lanza una excepción y no nos permite clonar ese objeto.

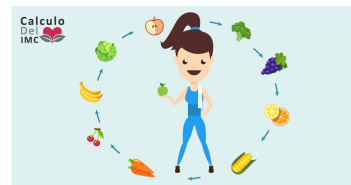
```
Mi nombre es: Ricardo Moya
No se puede clonar un objeto de la clase SoyUnico
java.lang.NullPointerException
    at Main.Main.main(Main.java:21)
```

En resumen, en este tutorial se ha puesto un ejemplo muy sencillo para ver el funcionamiento del patrón "**Singleton**" que es un patron de diseño muy utilizado (quizas el que más), aplicable en los casos en los que solo queremos tener un único objeto de la clase que implementamos.

Comparte esta entrada en:



all all all all



**Crea Tu Logo en 5 Minutos** **Map en Java, con ejemplos** **Cálculo del peso ideal**

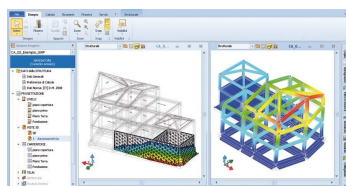
**Multitarea e H**  
**con ejemplos**  
**Runnable**

Anuncio Tailor Brands

jarroba.com

Anuncio Calculo Del IMC

jarroba.com



**Software EdiLus**

**Scraping en Java (JSoup),**  
**con ejemplos**

**Algoritmo de**  
**permutaciones sin**  
**repetición**

**Patrón Single**  
**Python, con e**

Anuncio ACCA software

jarroba.com

jarroba.com

jarroba.com



(<https://creativecommons.org/licenses/by-nc-sa/3.0/es/>)  
/userfeed/1401370112309)



(<https://www.safecreative.org>)

Patrón Singleton en Java, con ejemplos (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/>) por  
"www.jarroba.com" esta bajo una licencia Creative Commons

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de nuestra política de cookies (<http://jarroba.com>)

Reconocimiento No Comercial Compartir Igual 3.0 Unported License

Creado a partir de la obra en [www.jarroba.com](http://jarroba.com) (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/>)

(<http://jarroba.com/faq/>)

a.co A% rob Jav

## 13 thoughts on “Patrón Singleton en Java, con ejemplos”



**Gabriel Carrillo** dice:

25/10/2018 a las 01:35 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-224958>)

Muchas gracias me ayudo mucho y esta muy bien explicado



Responder



**Aleks** dice:

24/10/2018 a las 21:08 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-224940>)

Muuuuchisimas gracias me has salvado



Responder



**Miguel** dice:

19/05/2018 a las 01:37 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-211258>)

Muy clara la explicación y útil, mil gracias.



Responder



**Ella** dice:

12/10/2018 a las 12:24 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-223670>)

salame



Responder



**Roberto Barrientos** dice:

26/01/2018 a las 21:22 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-194776>)

si mando a llamar el método getInstance, para después mandar los setters con los valores esto no tendría ningún problema, cierto?



Responder



**Ramón [Admin Jarroba]** dice:

05/03/2018 a las 21:01 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-202883>)

No deberías tener ningún problema, pues con getInstance te devuelve el puntero al objeto entero. te podría dar problemas si devuelve Null, que no debiera.

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)



Responder

ACEPTAR



**Yo** dice:

24/07/2017 a las 04:12 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-173695>)

¡Explicación simple! Me ha sido útil. Muchas gracias.

 Responder



**glend** dice:

04/05/2017 a las 17:34 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-155093>)

Es aconsejable utilizar el patrón singleton para un sistema web?

creo que podría generar bloqueos si solo se usa una conexión.

 Responder



**Ramón [Admin Jarroba]** dice:

12/06/2017 a las 19:15 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-166172>)

El patrón Singleton es muy útil en ciertos casos, pero sí que pudiera haber problemas con la multitarea (si utilizas por ejemplo Spring), siempre puedes poner synchronized para evitar problemas pero esperas vas a tener. Dependerá de lo que necesites.

 Responder



**fas** dice:

08/01/2016 a las 19:05 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-30505>)

Asi como el clone no faltaria algo para el tema de multiples hilos accediendo?

 Responder



**Raúl** dice:

30/12/2015 a las 20:48 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-29935>)

Gracias por la explicacion!

 Responder



**Agustín** dice:

28/12/2015 a las 13:40 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-29704>)

Hola! Y feliz Navidad!

Uso de cookies

Entonces, en realidad si podríamos clonarlo pero estaría conceptualmente mal, no??  
Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

Saludos

Agustín

Responder



**Richard [Admin Jarroba]** dice:

31/12/2015 a las 08:16 (<https://jarroba.com/patron-singleton-en-java-con-ejemplos/#comment-29971>)

Hola Agustín.

La idea del patrón singleton es que solo haya un objeto de esa clase. Si lo clonamos obviamente habrá 2 o más objetos con lo cual no tendría sentido utilizar este patrón. No es que conceptualmente este mal, pero haciendo un clone no estarías utilizando este patrón. Normalmente cuando he utilizado este patrón no he implementado la excepción del clone ya que asumo que estoy utilizando ese patrón y no voy a clonar ese objeto, pero lo pongo en este tutorial por si acaso.

Saludos

Responder

## Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con \*

### Comentario

Nombre \*

Correo electrónico \*

Web

Publicar comentario

Este sitio usa Akismet para reducir el spam. Aprende cómo se procesan los datos de tus comentarios (<https://akismet.com/privacy/>).

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR



(/)

Síguenos en:



(<https://www.youtube.com/user/JarrobaWeb>)



(<https://twitter.com/JarrobaWeb>)



(<https://www.facebook.com/jarrobaWeb>)

(<https://github.com/jarroba?tab=repositories>)



(<https://plus.google.com/103352032205227460108/>)

Contacto: [jarrobaweb@gmail.com](mailto:jarrobaweb@gmail.com) (<mailto:jarrobaweb@gmail.com>)

---

[FAQ \(HTTPS://JARROBA.COM/FAQ/\)](https://jarroba.com/faq/)

---

[VISITAS \(HTTPS://JARROBA.COM/VISITAS/\)](https://jarroba.com/visitas/)

---

[GASTOS](#)

---

[MAPA DEL SITIO \(HTTPS://JARROBA.COM/SITEMAP\\_INDEX.XML\)](https://jarroba.com/sitemap_index.xml)

---

[HERRAMIENTAS \(HTTPS://JARROBA.COM/TOOLS/\)](https://jarroba.com/tools/)

---

Copyright © 2019 Jarroba.com (<https://www.jarroba.com>) - Todos los derechos reservados

*By Reimon & Richard*

---

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

---