

(/)

Comprobación de cadenas vacías o en blanco en Java

Última modificación: 7 de julio de 2019

por baeldung (<https://www.baeldung.com/author/baeldung/>)
(<https://www.baeldung.com/author/baeldung/>)

Java (<https://www.baeldung.com/category/java/>) +

Acabo de anunciar el nuevo curso *Learn Spring*, enfocado en los fundamentos de Spring 5 y Spring Boot 2:

>> VISITE EL CURSO (</ls-course-start>)

1. Introducción

En este tutorial, veremos algunas formas de verificar cadenas vacías o en blanco en Java. Tenemos algunos enfoques de idioma nativo, así como un par de bibliotecas.

2. Vacío contra blanco

Por supuesto, es bastante común saber cuándo una cadena está vacía o en blanco, pero asegurémonos de estar en la misma página con nuestras definiciones.

Consideramos que una cadena está *vacía* si es *nula* o una cadena sin longitud *alguna*. Si una cadena solo consta de espacios en blanco solamente, entonces lo llamamos en *blanco*.

Para Java, los espacios en blanco son caracteres como espacios, pestañas, etc. Echa un vistazo a *Character.isWhitespace* (<https://docs.oracle.com/javase/8/docs/api/java/lang/Character.html#isWhitespace-char->) para ver ejemplos.

3. Cuerdas vacías

3.1. Con Java 6 y superior

Si estamos al menos en Java 6, la forma más sencilla de verificar si hay una cadena *vacía* es *String # isEmpty* (<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#isEmpty-->):

```
1 | boolean isEmptyString(String string) {  
2 |     return string == null || string.isEmpty();  
3 | }
```

Para que también sea seguro para la nulidad, necesitamos agregar un cheque adicional.

3.2. Con Java 5 y Abajo

String # isEmpty se introdujo con Java 6. Para Java 5 y versiones anteriores, podemos usar *String # length* en su lugar.

```
1 | boolean isEmptyString(String string) {  
2 |     return string == null || string.length() == 0;  
3 | }
```

De hecho, ***String # isEmpty*** es solo un acceso directo a ***String # length***.

4. Cuerdas en blanco

Tanto *String # isEmpty* como *String # length* pueden usarse para verificar si hay cadenas vacías.

Si también deseamos detectar cadenas en *blanco*, podemos lograrlo con la ayuda de *String # trim*. Se **eliminarán todos los espacios en blanco iniciales y finales antes de realizar la comprobación**.

```
1 | boolean isBlankString(String string) {  
2 |     return string == null || string.trim().isEmpty();  
3 | }
```

Para ser precisos, *el ajuste String #* eliminará todos los caracteres iniciales y finales con un código Unicode menor o igual que U + 0020 (https://en.wikipedia.org/wiki/List_of_Unicode_characters#Control_codes).

Y también recuerda que los *String*s son inmutables, por lo que el *recorte de llamada* no cambiará realmente el string subyacente.

5. Validación de frijoles

Otra forma de verificar si hay cadenas en *blanco* es mediante expresiones regulares. Esto es útil, por ejemplo, con la validación de Java Bean (<https://www.baeldung.com/javax-validation>):

```
1 | @Pattern(regex = "\\A(?:!\\s*\\Z).+")  
2 | String someString;
```

La expresión regular dada garantiza que las cadenas vacías o en blanco no se validarán.

6. Con Apache Commons

Si está bien agregar dependencias, podemos usar Apache Commons Lang (<https://commons.apache.org/proper/commons-lang/>). Esto tiene una gran cantidad de ayudantes para Java.

Si usamos Maven, necesitamos agregar la dependencia *commons-lang3* (<https://search.maven.org/search?q=g:org.apache.commons%20AND%20a:commons-lang3>) a nuestro pom:

```
1 <dependency>
2     <groupId>org.apache.commons</groupId>
3     <artifactId>commons-lang3</artifactId>
4 </dependency>
```

Entre otras cosas, esto nos da *StringUtils* (<https://commons.apache.org/proper/commons-lang/apidocs/org/apache/commons/lang3/StringUtils.html>) .

Esta clase viene con métodos como *isEmpty* (<https://commons.apache.org/proper/commons-lang/apidocs/org/apache/commons/lang3/StringUtils.html#isEmpty-java.lang.CharSequence->) , *isBlank* (<https://commons.apache.org/proper/commons-lang/apidocs/org/apache/commons/lang3/StringUtils.html#isBlank-java.lang.CharSequence->) , etc.

```
1 StringUtils.isBlank(string)
```

Esta llamada hace lo mismo que nuestro propio método *isBlankString* . Es nulo seguro y también verifica espacios en blanco.

7. Con guayaba

Otra biblioteca conocida que trae ciertas utilidades relacionadas con cadenas es la Guava (<https://github.com/google/guava>) de Google . A partir de la versión 23.1, hay dos tipos de guayaba: *android* y *jre* . El sabor de Android se dirige a Android y Java 7, mientras que el sabor de JRE se aplica a Java 8.

Si no estamos apuntando a Android, solo podemos agregar el sabor JRE (<https://search.maven.org/artifact/com.google.guava/guava/28.0-jre/jar>) a nuestro pom:

```
1 <dependency>
2     <groupId>com.google.guava</groupId>
3     <artifactId>guava</artifactId>
4     <version>28.0-jre</version>
5 </dependency>
```

La clase Guavas Strings viene con un método *Strings.isNullOrEmpty* (<https://google.github.io/guava/releases/27.1-jre/api/docs/com/google/common/base/Strings.html#isNullOrEmpty-java.lang.String->) :

```
1 | Strings.isNullOrEmpty(string)
```

Comprueba si una cadena dada es nula o está vacía, **pero no buscará cadenas de solo espacios en blanco** .

8. Conclusión

Hay varias formas de verificar si una cadena está vacía o no. A menudo, también queremos comprobar si una cadena está en blanco, lo que significa que solo está formada por caracteres de espacio en blanco.

La forma más conveniente es usar Apache Commons Lang, que proporciona ayudantes como *StringUtils.isBlank* . Si queremos mantenernos en Java simple, podemos usar una combinación de *String # trim* con *String # isEmpty* o *String # length* . Para la Validación de Bean, se pueden usar expresiones regulares en su lugar.

Asegúrate de revisar todas estas muestras en GitHub (<https://github.com/eugenp/tutorials/tree/master/java-strings-2>) .

Acabo de anunciar el nuevo curso *Learn Spring*, enfocado en los fundamentos de Spring 5 y Spring Boot 2:

>> VISITE EL CURSO (/ls-course-end)



¿Aprender a "construir su API con Spring"?

Enter your email address

>> Consigue el eBook

Deja una respuesta



Start the discussion...

✉ Suscribir ▼

LAS CATEGORÍAS

[PRIMAVERA \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/SPRING/\)](https://www.baeldung.com/category/spring/)
[DESCANSO \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/REST/\)](https://www.baeldung.com/category/rest/)
[JAVA \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/JAVA/\)](https://www.baeldung.com/category/java/)
[SEGURIDAD \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/\)](https://www.baeldung.com/category/security-2/)
[PERSISTENCIA \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/\)](https://www.baeldung.com/category/persistence/)
[JACKSON \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/JSON/JACKSON/\)](https://www.baeldung.com/category/json/jackson/)
[CLIENTE HTTP \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/HTTP/\)](https://www.baeldung.com/category/http/)
[KOTLIN \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/KOTLIN/\)](https://www.baeldung.com/category/kotlin/)

SERIE

[TUTORIAL DE JAVA "BACK TO BASICS" \(/JAVA-TUTORIAL\)](#)
[JACKSON JSON TUTORIAL \(/JACKSON\)](#)
[TUTORIAL HTTPCLIENT 4 \(/HTTPCLIENT-GUIDE\)](#)
[DESCANSO CON TUTORIAL DE PRIMAVERA \(/REST-WITH-SPRING-SERIES\)](#)
[TUTORIAL DE PERSISTENCIA DE PRIMAVERA \(/PERSISTENCE-WITH-SPRING-SERIES\)](#)
[SEGURIDAD CON SPRING \(/SECURITY-SPRING\)](#)

ACERCA DE

[ACERCA DE BAELDUNG \(/ABOUT\)](#)
[LOS CURSOS \(HTTPS://COURSES.BAELDUNG.COM\)](https://courses.baeldung.com)
[TRABAJO DE CONSULTORÍA \(/CONSULTING\)](#)
[META BAELDUNG \(HTTP://META.BAELDUNG.COM/\)](http://meta.baeldung.com/)
[EL ARCHIVO COMPLETO \(/FULL_ARCHIVE\)](#)
[ESCRIBIR PARA BAELDUNG \(/CONTRIBUTION-GUIDELINES\)](#)
[EDITORES \(/EDITORS\)](#)
[NUESTROS COMPAÑEROS \(/PARTNERS\)](#)
[PUBLICIDAD EN BAELDUNG \(/ADVERTISE\)](#)

[TÉRMINOS DE SERVICIO \(/TERMS-OF-SERVICE\)](#)
[POLÍTICA DE PRIVACIDAD \(/PRIVACY-POLICY\)](#)
[INFORMACIÓN DE LA COMPAÑÍA \(/BAELDUNG-COMPANY-INFO\)](#)

CONTACTO (/CONTACT)