


( / )



### Especialistas en Hosting JAVA

Publicar en Internet tu aplicación web desarrollada en JAVA sólo te llevará unos segundos.

 Alojamiento ANW® JAVA

VER MÁS

# Hacer capturas de pantalla con Java

Última modificación: 31 de julio de 2020

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*, centrado en los fundamentos de Spring 5 y Spring Boot 2:

>> VER EL CURSO (/ls-course-start)

## 1. Introducción

En este tutorial, veremos algunas formas diferentes de tomar una captura de pantalla en Java.

## 2. Tomar una captura de pantalla con un *robot*



En nuestro primer ejemplo, vamos a tomar una captura de pantalla de la pantalla principal.

Para eso, usaremos el método `createScreenCapture()` de la clase `Robot`. Toma un `Rectangle` como parámetro que establece los límites para la captura de pantalla y devuelve un objeto `BufferedImage`. El `BufferedImage` se puede utilizar además para crear un archivo de imagen:

Okay



```

1  @Test
2  public void givenMainScreen_whenTakeScreenshot_thenSaveToFile() throws Exception {
3      Rectangle screenRect = new Rectangle(Toolkit.getDefaultToolkit().getScreenSize());
4      BufferedImage capture = new Robot().createScreenCapture(screenRect);
5
6      File imageFile = new File("single-screen.bmp");
7      ImageIO.write(capture, "bmp", imageFile);
8      assertTrue(imageFile.exists());
9  }

```

Se puede acceder a las dimensiones de la pantalla a través de la clase *Toolkit* mediante su método *getScreenSize()*. En sistemas con varias pantallas, la pantalla principal se utiliza de forma predeterminada.

Después de capturar la pantalla en *BufferedImage*, podemos escribirla en el archivo con *ImageIO.write()*. Para hacerlo, necesitaremos dos parámetros adicionales. El formato de imagen y el archivo de imagen en sí. En nuestro ejemplo, estamos usando *bmp*, pero a otros les gusta *png*, *jpg* o *gif* también están disponibles.

### 3. Tomar una captura de pantalla de varias pantallas

También es posible tomar una captura de pantalla de varias pantallas a la vez. Al igual que en el ejemplo anterior, podemos usar el método *createScreenCapture()* de la clase *Robot*. Pero esta vez los límites de la captura de pantalla deben cubrir todas las pantallas requeridas.

Para obtener todas las pantallas, usaremos la clase *GraphicsEnvironment* y su método *getScreenDevices()*.

A continuación, buscaremos el límite de cada pantalla individual y crearemos un *Rectángulo* que se ajustará a todas:

```

1  @Test
2  public void givenMultipleScreens_whenTakeScreenshot_thenSaveToFile() throws Exception {
3      GraphicsEnvironment ge = GraphicsEnvironment.getLocalGraphicsEnvironment();
4      GraphicsDevice[] screens = ge.getScreenDevices();
5
6      Rectangle allScreenBounds = new Rectangle();
7      for (GraphicsDevice screen : screens) {
8          Rectangle screenBounds = screen.getDefaultConfiguration().getBounds();
9          allScreenBounds.width += screenBounds.width;
10         allScreenBounds.height = Math.max(allScreenBounds.height, screenBounds.height);
11     }
12
13     BufferedImage capture = new Robot().createScreenCapture(allScreenBounds);
14     File imageFile = new File("all-screens.bmp");
15     ImageIO.write(capture, "bmp", imageFile);
16     assertTrue(imageFile.exists());
17 }

```



Mientras iteramos sobre las pantallas, siempre sumamos los anchos y elegimos solo una altura máxima, ya que las pantallas se concatenarán horizontalmente.

Yendo más lejos, necesitamos guardar la imagen de captura de pantalla. Como en el ejemplo anterior, podemos usar el método *ImageIO.write()*.

### 4. Tomar una captura de pantalla de un componente de GUI dado

También podemos tomar una captura de pantalla de un componente de la interfaz de usuario determinado.

Se puede acceder fácilmente a las dimensiones a través del método *getBounds()* ya que cada

Usamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la [Política de privacidad y cookies completa \(/privacy-policy\)](#)

Okay



En este caso, no vamos a utilizar la API de *robot*. En su lugar, usaremos el método *paint()* de la clase *Component* que dibujará el contenido directamente en *BufferedImage*:

```
1  @Test
2  public void givenComponent_whenTakeScreenshot_thenSaveToFile(Component component) throws Exception
3  {
4      Rectangle componentRect = component.getBounds();
5      BufferedImage bufferedImage = new BufferedImage(componentRect.width, componentRect.height,
6      BufferedImage.TYPE_INT_ARGB);
7      component.paint(bufferedImage.getGraphics());
8
9      File imageFile = new File("component-screenshot.bmp");
10     ImageIO.write(bufferedImage, "bmp", imageFile);
11     assertTrue(imageFile.exists());
12 }
```

Después de obtener el enlace del componente, necesitamos crear la imagen *Buffered*. Para esto, necesitamos el ancho, alto y tipo de imagen. En este caso, estamos usando *BufferedImage.TYPE\_INT\_ARGB* que se refiere a una imagen en color de 8 bits.

Luego avanzamos para invocar el método *paint()* para completar la *Imagen Buffered* y, al igual que en los ejemplos anteriores, la guardamos en un archivo con el método *ImageIO.write()*.

## 5. Conclusión

En este tutorial, hemos aprendido varias formas de realizar capturas de pantalla con Java.

Como siempre, el código fuente con todos los ejemplos de este tutorial está disponible en GitHub (<https://github.com/eugenp/tutorials/tree/master/core-java-modules/core-java-os>).



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

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



¿Está aprendiendo a "construir su API con Spring"?

>> Obtenga el eBook



Iniciar sesión ([https://www.baeldung.com/wp-login.php?redirect\\_to=https%3A%2F%2Fwww.baeldung.com%2Fjava-taking-screenshots](https://www.baeldung.com/wp-login.php?redirect_to=https%3A%2F%2Fwww.baeldung.com%2Fjava-taking-screenshots))



***Be the First to Comment!***

**B I U**



Usamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la [Política de privacidad y cookies completa](https://www.baeldung.com/privacy-policy) ([/privacy-policy](https://www.baeldung.com/privacy-policy))

Okay



 **ezoic** (<https://www.ezoic.com/what-is-ezoic/>)

Quéjate de este anun

## 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/))  
LADO DEL 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 'VOLVER A LO BÁSICO' ([/JAVA-TUTORIAL](#))  
TUTORIAL DE JACKSON JSON ([/JACKSON](#))  
TUTORIAL DE HTTPCLIENT 4 ([/HTTPCLIENT-GUIDE](#))  
DESCANSO CON SPRING TUTORIAL ([/REST-WITH-SPRING-SERIES](#))  
TUTORIAL DE PERSISTENCIA DE PRIMAVERA ([/PERSISTENCE-WITH-SPRING-SERIES](#))  
SEGURIDAD CON SPRING ([/SECURITY-SPRING](#))

## ACERCA DE

SOBRE BAELDUNG ([/ABOUT](#))  
LOS CURSOS ([HTTPS://COURSES.BAELDUNG.COM](https://courses.baeldung.com))  
TRABAJOS ([/TAG/ACTIVE-JOB/](#))  
EL ARCHIVO COMPLETO ([/FULL\\_ARCHIVE](#))  
ESCRIBE PARA BAELDUNG ([/CONTRIBUTION-GUIDELINES](#))  
EDITORES ([/EDITORS](#))  
NUESTROS COMPAÑEROS ([/PARTNERS](#))  
ANUNCIE EN BAELDUNG ([/ADVERTISE](#))

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

Usamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la [Política de privacidad y cookies completa \(/privacy-policy\)](#)

Okay



Usamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la [Política de privacidad y cookies completa \(/privacy-policy\)](#)

Okay