

# Enriqueciendo el ecosistema de herramientas de código abierto para el desarrollo de pruebas con Selenium

Software Freedom Day  
23/09/2019

Boni García

 [boni.garcia@urjc.es](mailto:boni.garcia@urjc.es)  <http://bonigarcia.github.io/>

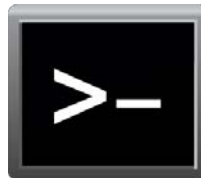
 [@boni\\_gg](https://twitter.com/boni_gg)  <https://github.com/bonigarcia>

# Contenidos

WebDriverManager

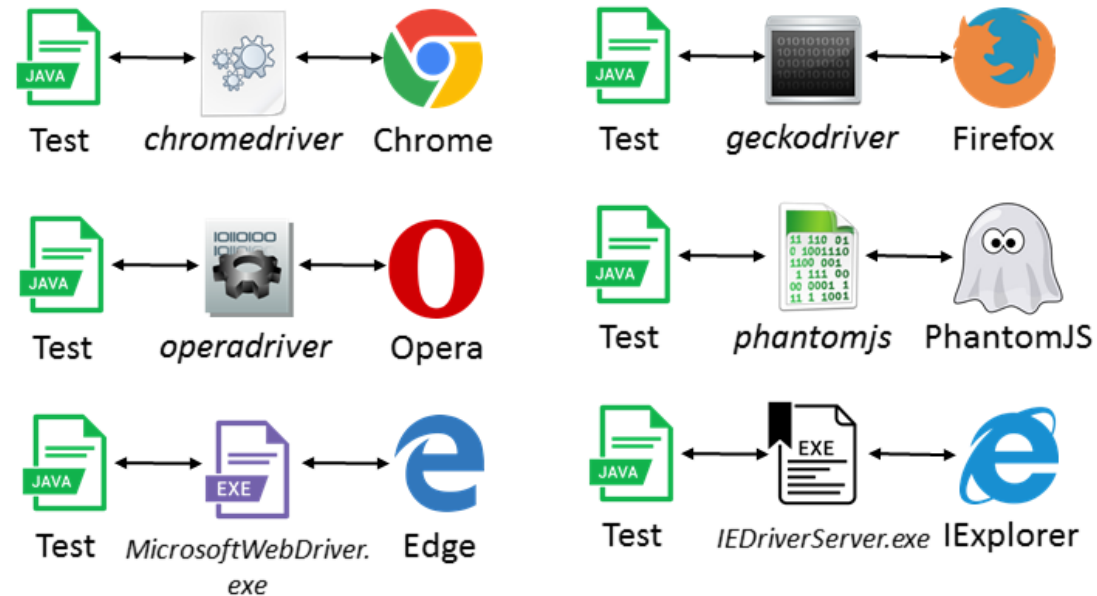


Selenium-Jupiter



# WebDriverManager - Introducción

- **Selenium WebDriver** permite controlar navegadores web de forma programática (pruebas *end-to-end*)



```
System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");  
System.setProperty("webdriver.opera.driver", "/path/to/operadriver");  
System.setProperty("webdriver.ie.driver", "C:/path/to/IEDriverServer.exe");  
System.setProperty("webdriver.edge.driver", "C:/path/to/MicrosoftWebDriver.exe");  
System.setProperty("phantomjs.binary.path", "/path/to/phantomjs");  
System.setProperty("webdriver.gecko.driver", "/path/to/geckodriver");
```



# WebDriverManager - Problema





# WebDriverManager - Solución

- **WebDriverManager** es una librería Java que permite automatizar la gestión de los binarios requeridos por Selenium WebDriver (chromedriver, geckodriver, etc.)

Fork me on GitHub

```
dependencies {  
    testCompile("io.github.bonigarcia:webdrivermanager:3.7.1")  
}
```



```
<dependency>  
    <groupId>io.github.bonigarcia</groupId>  
    <artifactId>webdrivermanager</artifactId>  
    <version>3.7.1</version>  
    <scope>test</scope>  
</dependency>
```

**Maven™**

```
WebDriverManager.chromedriver().setup();  
WebDriverManager.firefoxdriver().setup();  
WebDriverManager.operadriver().setup();  
WebDriverManager.edgedriver().setup();  
WebDriverManager.iedriver().setup();  
WebDriverManager.phantomjs().setup();
```



<https://github.com/bonigarcia/webdrivermanager>

# WebDriverManager - Uso básico

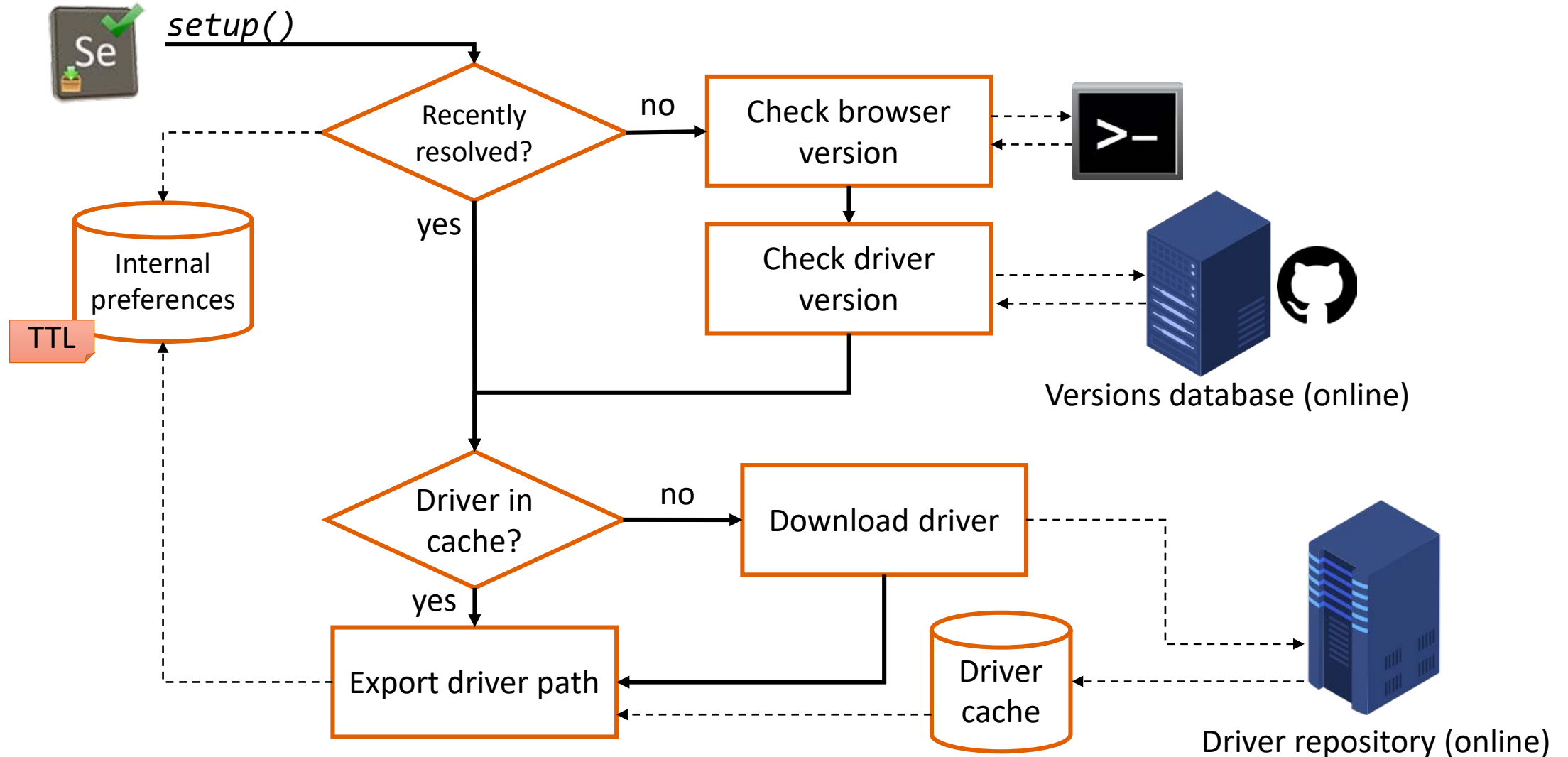
```
public class ChromeTest {  
  
    private WebDriver driver;  
  
    @BeforeClass  
    public static void setupClass() {  
        WebDriverManager.chromedriver().setup();  
    }  
  
    @Before  
    public void setupTest() {  
        driver = new ChromeDriver();  
    }  
  
    @After  
    public void teardown() {  
        if (driver != null) {  
            driver.quit();  
        }  
    }  
  
    @Test  
    public void test() {  
        // Your test code here  
    }  
}
```



```
public class ChromeTest {  
  
    private WebDriver driver;  
  
    @BeforeClass  
    public static void setupClass() {  
        WebDriverManager.firefoxdriver().setup();  
    }  
  
    @Before  
    public void setupTest() {  
        driver = new FirefoxDriver();  
    }  
  
    @After  
    public void teardown() {  
        if (driver != null) {  
            driver.quit();  
        }  
    }  
  
    @Test  
    public void test() {  
        // Your test code here  
    }  
}
```



# WebDriverManager - Algoritmo



# WebDriverManager - Más allá de Java

- WebDriverManager puede ser usado también como:

## 1. Herramienta CLI (Command Line Interface) :

```
> java -jar webdrivermanager-3.7.1-fat.jar chrome
[INFO] Using WebDriverManager to resolve chrome
[INFO] Using chromedriver 76.0.3809.126 (since Google Chrome 76 is installed in your machine)
[INFO] Reading https://chromedriver.storage.googleapis.com/ to seek chromedriver
[INFO] Downloading https://chromedriver.storage.googleapis.com/76.0.3809.126/chromedriver_win32.zip
[INFO] Extracting binary from compressed file chromedriver_win32.zip
[INFO] Resulting binary C:\Users\boni\Downloads\chromedriver.exe
```

## 2. Como servidor (API REST-like):

```
> java -jar webdrivermanager-3.7.1-fat.jar server
[INFO] WebDriverManager server listening on port 4041
```

Ejemplo de peticiones a WebDriverManager Server:

<http://localhost:4041/firefoxdriver>

<http://localhost:4041/chromedriver?chromeDriverVersion=2.40>



# WebDriverManager - Conclusiones

- WebDriverManager resuelve un problema real de un ámbito específico de la comunidad de *testing*
  - Estadísticas Maven Central: 345K descargas, 33K IP únicas (agosto 2019)
  - Estadísticas GitHub: 947 ★, usado por 11K proyectos
  - Usado en proyectos relevantes como Appium y Selenide
  - Referido en un post del blog de Martin Fowler
- Han surgido herramientas similares en otros lenguajes:
  - webdriver-manager (Node.js): <https://github.com/angular/webdriver-manager>
  - webdriver\_manager (Python): [https://github.com/jeffnyman/webdriver\\_manager](https://github.com/jeffnyman/webdriver_manager)
  - WebDriverManager.Net (.Net): <https://github.com/rosolko/WebDriverManager.Net>
  - Webdrivers Gem (Ruby): <https://github.com/titusfortner/webdrivers>

# Contenidos

WebDriverManager



Selenium-Jupiter



# Selenium-Jupiter - Introducción

- **JUnit 5** es la última versión del popular framework de pruebas JUnit
- Proporciona un modelo de programación y extensión llamado **Jupiter**
- El modelo de extensión es muy adecuado para añadir nuevas funcionales (por ejemplo, Selenium)



<https://junit.org/junit5/docs/current/user-guide/>

```
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
```

```
class BasicJUnit5Test {

    @BeforeAll
    static void setupAll() {
        // setup all tests
    }

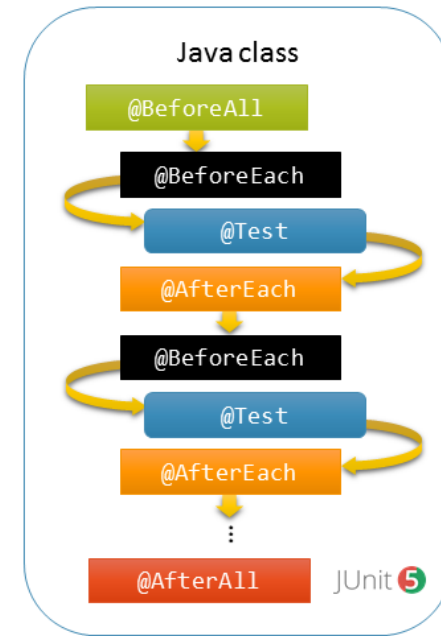
    @BeforeEach
    void setup() {
        // setup each test
    }

    @Test
    void test() {
        // exercise and verify SUT
    }

    @AfterEach
    void teardown() {
        // teardown each test
    }

    @AfterAll
    static void teardownAll() {
        // teardown all tests
    }

}
```



# Selenium-Jupiter - Objetivos

- **Selenium-Jupiter** es una extensión JUnit 5 que permite el desarrollo de casos de pruebas avanzadas con Selenium. Los objetivos específicos que se persiguen son:

- Permitir crear tests de forma sencilla (reducir el *boilerplate*)
- Eliminar procesos manuales
- Integración sencilla con Docker (para usar navegadores web y dispositivos Android en contenedores)
- Ofrecer herramientas avanzadas para el desarrollo de pruebas web y móviles



<https://github.com/bonigarcia/selenium-jupiter>

Fork me on GitHub

# Selenium-Jupiter - Uso básico

- **Selenium-Jupiter** está basado en la inyección de dependencias de JUnit 5

```
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>selenium-jupiter</artifactId>
  <version>3.3.1</version>
  <scope>test</scope>
</dependency>
```

**Maven™**

```
dependencies {
  testCompile("io.github.bonigarcia:selenium-jupiter:3.3.1")
}
```



Se pueden declarar parámetros de los siguientes tipos:  
ChromeDriver, FirefoxDriver, OperaDriver,  
SafariDriver, EdgeDriver,  
InternetExplorerDriver, HtmlUnitDriver,  
PhantomJSDriver, AppiumDriver,  
SelenideDriver

```
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.github.bonigarcia.SeleniumExtension;

@ExtendWith(SeleniumExtension.class)
class ChromeAndFirefoxJupiterTest {

    @Test
    void testWithOneChrome(ChromeDriver chromeDriver) {
        // Use Chrome in this test
    }

    @Test
    void testWithFirefox(FirefoxDriver firefoxDriver) {
        // Use Firefox in this test
    }

    @Test
    void testWithChromeAndFirefox(ChromeDriver chromeDriver,
        FirefoxDriver firefoxDriver) {
        // Use Chrome and Firefox in this test
    }




}
```

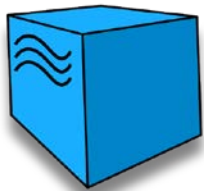




# Selenium-Jupiter - Docker

- Selenium-Jupiter permite usar navegadores web y dispositivos Android en contenedores **Docker**:

- Chrome, Firefox, y Opera: 
  - Versiones estables de las imágenes Docker mantenidas por **Aerokube**
  - Versiones beta e inestable (Chrome y Firefox) mantenidas por **ElasTest**
- Edge and Internet Explorer: 
  - Debido a la licencia de Windows, estas imágenes no se pueden alojar en Docker Hub
  - No obstante, se pueden construir usando un tutorial proporcionado por [Aerokube](#)
- Dispositivos Android: 
  - Imágenes Docker mantenidas en el proyecto **docker-android** (por Budi Utomo)



# Selenium-Jupiter - Docker

- Ejemplo de uso de Docker:

```
import static io.github.bonigarcia.BrowserType.CHROME;
import static io.github.bonigarcia.BrowserType.FIREFOX;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.openqa.selenium.remote.RemoteWebDriver;

import io.github.bonigarcia.DockerBrowser;
import io.github.bonigarcia.SeleniumExtension;

@ExtendWith(SeleniumExtension.class)
class DockerChromeJupiterTest {

    @Test
    void testChromeStable(@DockerBrowser(type = CHROME) RemoteWebDriver driver) {
        // test
    }

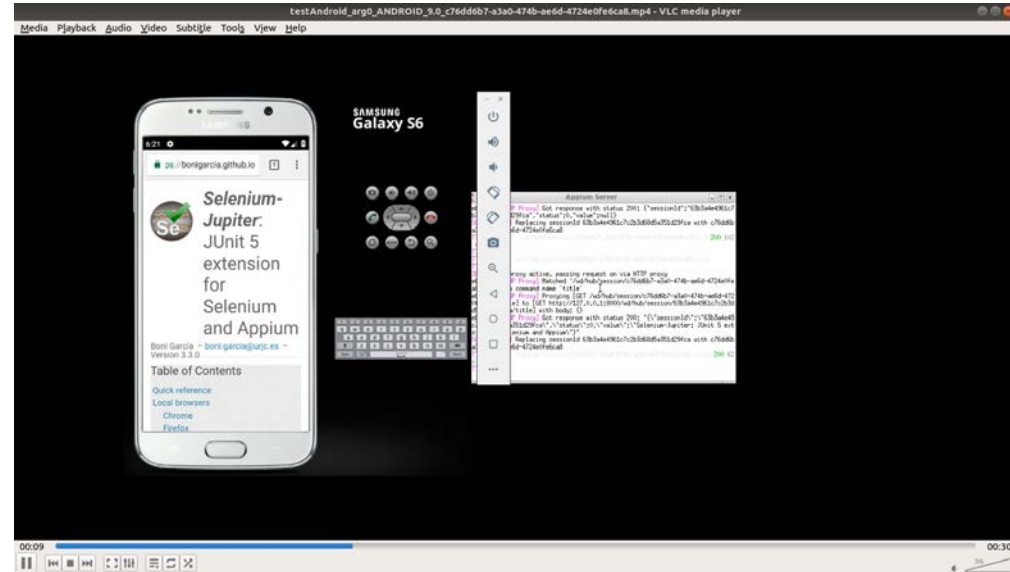
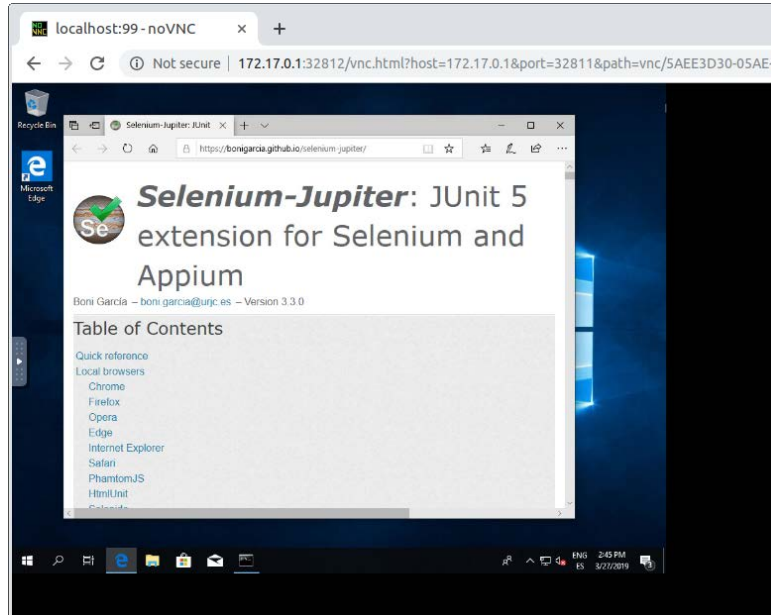
    @Test
    void testFirefoxBeta(@DockerBrowser(type = FIREFOX, version = "beta")
        RemoteWebDriver driver) {
        // test
    }
}
```

Tipo soportados: **CHROME**,  
**FIREFOX**, **OPERA**, **EDGE**,  
**IEXPLORER** and **ANDROID**

Si no especificamos el parámetro *version*, se usa la última versión disponible en Docker Hub. Se puede especificar una versión fija así como los valores **latest**, **latest-\***, **beta**, and **unstable**

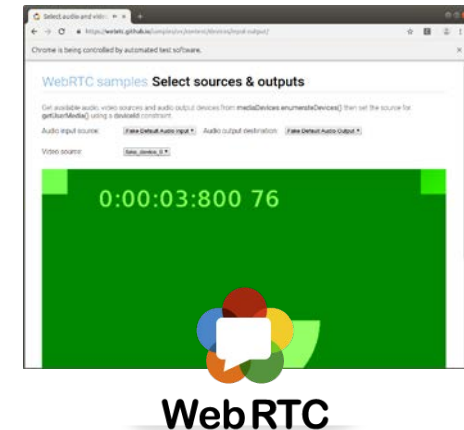
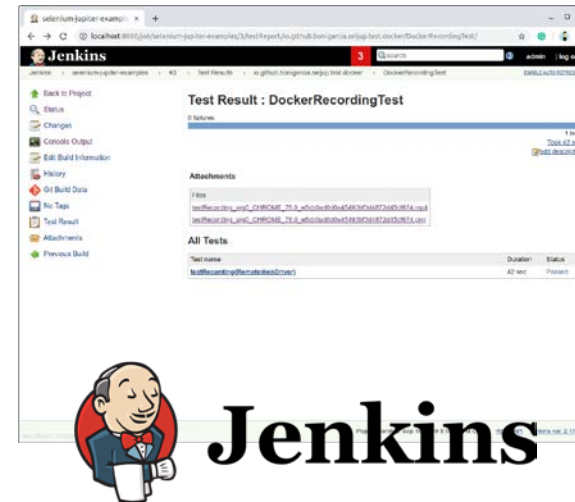
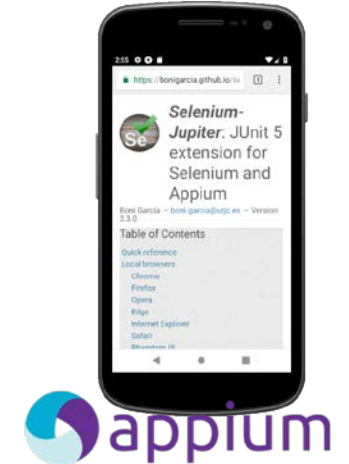
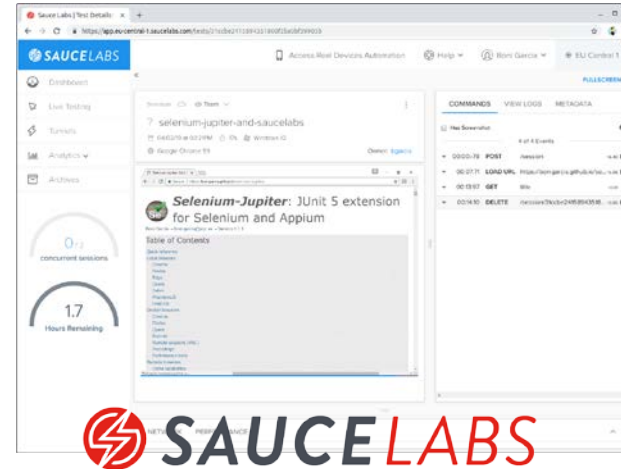
# Selenium-Jupiter - Docker

- El uso de Docker proporciona capacidades adicionales:
  - **Accesos remoto** a través de noVNC
  - **Grabación** de la sesión
  - Pruebas de **rendimiento**
  - Uso de diferentes versiones y dispositivos **Android**



# Selenium-Jupiter - Otros usos

- Uso de plantillas de pruebas
- Uso de navegadores remotos
- Declaración de opciones Selenium
- Integración con Jenkins
- Integración con Genymotion
- Gestión configurable de pantallazos al finalizar los tests
- Control avanzado de contenedores Docker
- Driver genérico configurable
- Configuración a través de código, propiedades JVM, o variables de entorno
- ...



# Selenium-Jupiter - Más allá de Java

- Selenium-Jupiter puede ser usado también como:
  1. Herramienta CLI (permite acceder de manera manual a navegadores a través de **VNC**) :

```
$ java -jar selenium-jupiter-3.3.1-fat.jar chrome unstable  
[INFO] Using Selenium-Jupiter to execute chrome unstable in Docker  
...
```

2. Como servidor (Selenium-Jupiter se convierte en un **Selenium Server**):

```
$ java -jar webdrivermanager-3.3.1-fat.jar server  
[INFO] Selenium-Jupiter server listening on http://localhost:4042/wd/hub  
...
```



# Selenium-Jupiter - Conclusiones

- Selenium-Jupiter propone un modelo de programación de pruebas Selenium avanzadas basado en JUnit 5
- Actualmente su uso no es muy extensivo:
  - Estadísticas Maven Central: 2K descargas, 680 IP únicas (agosto 2019)
  - Estadísticas GitHub: 120 ★, usado por 130 proyectos
- En el futuro próximo va a seguir evolucionado (*roadmap*):
  - Aumentar la escalabilidad de las pruebas de rendimiento, quizá usando Kubernetes
  - Mejorar las plantillas de test (por ejemplo, especificando opciones en formato JSON)
  - Recopilación de la consola de browsers (log JavaScript)

# Enriqueciendo el ecosistema de herramientas de código abierto para el desarrollo de pruebas con Selenium

¡Muchas gracias!  
Q&A

Boni García



[boni.garcia@urjc.es](mailto:boni.garcia@urjc.es)



<http://bonigarcia.github.io/>



[@boni\\_gg](https://twitter.com/boni_gg)



<https://github.com/bonigarcia>