



# Atam

Automatización de Pruebas Funcionales

Sesiones 3 y 4

18/05/2022

CEIIST -spa- v1.0

panel.es

Panel Sistemas Informáticos, S.L.

Consultoría, servicios y soluciones TI.



SESIÓN 3

## Introducción

Beneficios VS Problemas de la Automatización

Principios de una Automatización Eficiente

SESIÓN 4

## Pruebas de Integración

Postman

SoapUI

Rest Assured

## Pruebas E2E

Selenium

Cypress

Appium

## Pruebas de Rendimiento

JMeter

k6.io



# Introducción

# *BENEFICIOS vs PROBLEMAS*

## *de la Automatización*



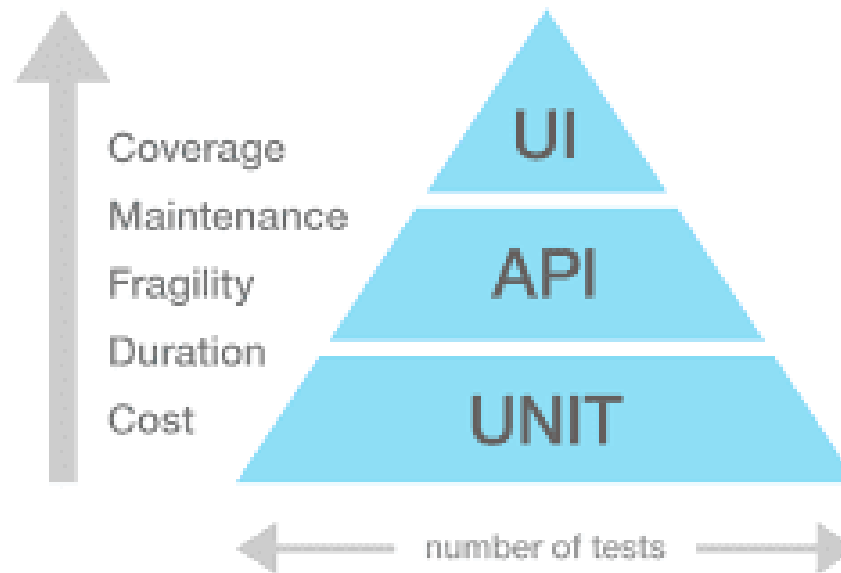
- ☐ Feedback temprano
- ☐ T2M más rápido
- ☐ Incrementa eficiencia equipo QA
- ☐ El equipo humano dedica su esfuerzo en tareas de mayor valor



- ☐ Datos de prueba
- ☐ Mantenimiento
- ☐ El control de la automatización es humano

- ❑ Inversión de esfuerzo en distintos niveles de prueba

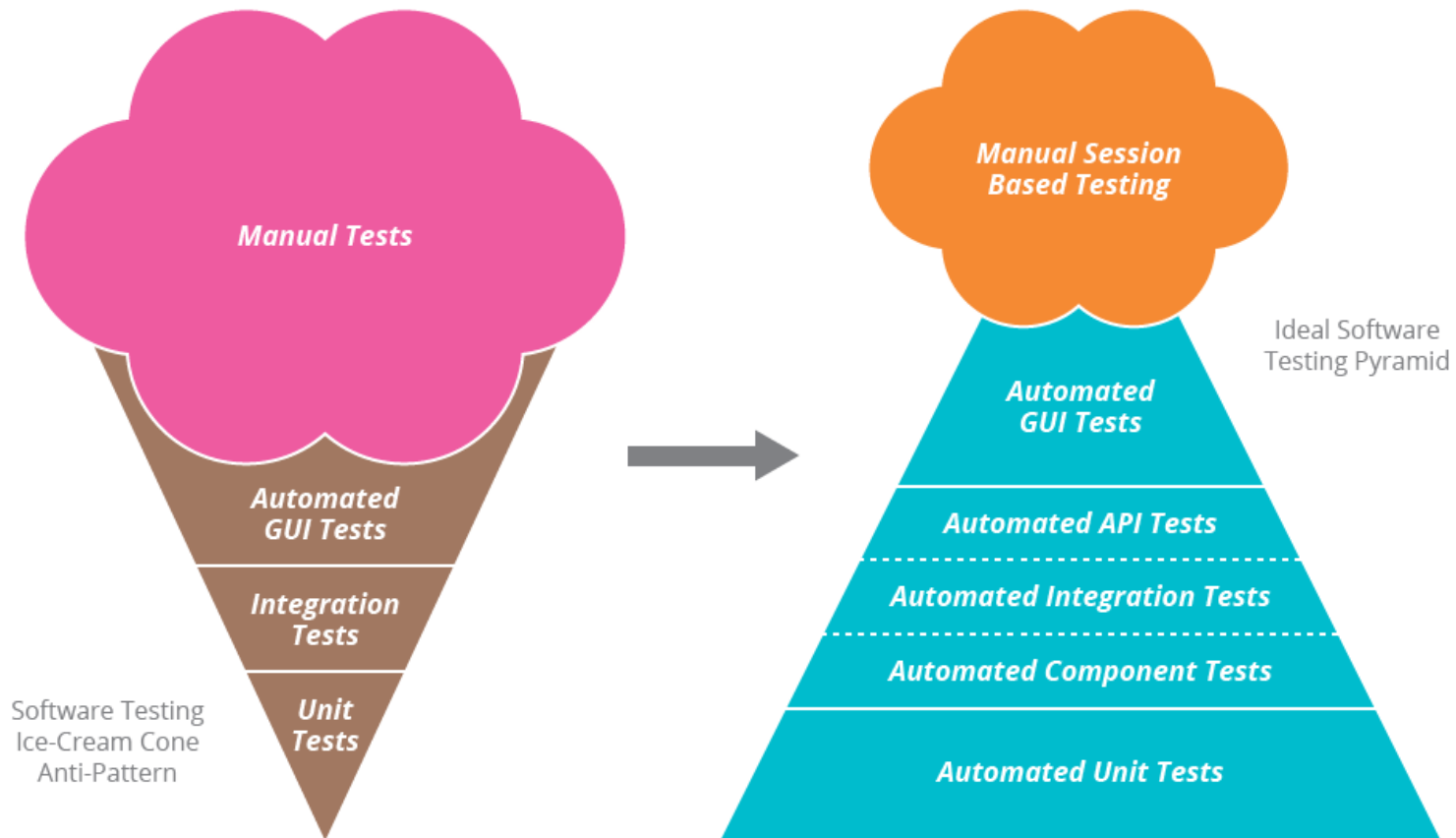
## The Automation Pyramid



\* Mike Cohn

# Principios de una Automatización EFICIENTE

❑ Pero en ocasiones ...



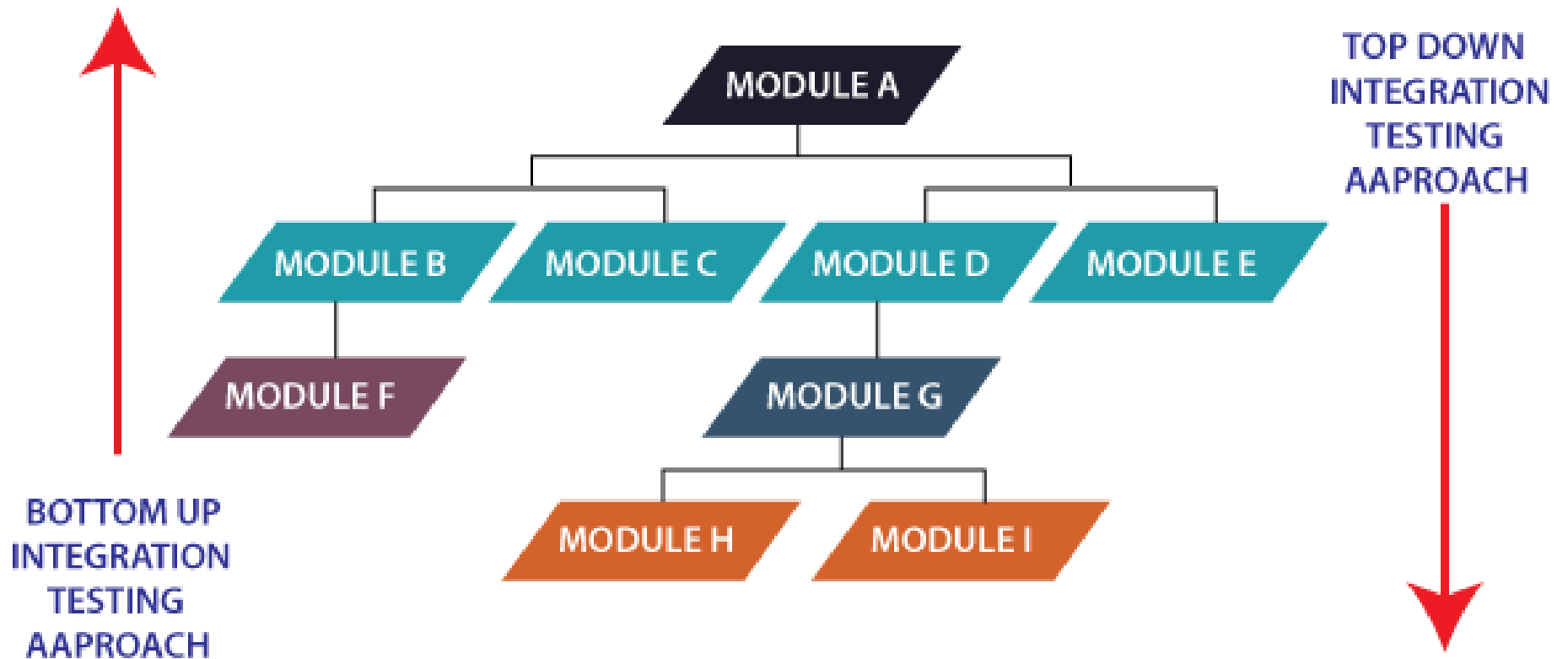


# Pruebas de Integración

- ❑ El SUT es un conjunto de subsistemas de la aplicación interactuando entre sí
  - El objetivo consiste en verificar que la comunicación entre los subsistemas es correcta
  - Habitualmente requiere la comunicación de diversos sistemas mediante protocolos de red o el uso de librerías de terceros
  - Cuanto menor sea el número de unidades a probar, más fácil será determinar el origen del error si el test falla



# Pruebas de integración



## Pruebas de Integración



POSTMAN



SoapUI



**REST-assured**

- ☐ Orientado a consumo de API REST
- ☐ Interfaz de usuario actual e intuitiva
- ☐ Facilita el trabajo colaborativo
- ☐ Estructura y gestión de información [colecciones, entornos y datos]
- ☐ Integración con otras herramientas / extensiones
- ☐ Permite ejecutar scripts JS antes / después de cada petición
- ☐ Curva de aprendizaje reducida



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**

- ☐ Orientado a SOA. Proyectos SOAP, REST y genéricos
- ☐ Interfaz de usuario desfasada y poco intuitiva
- ☐ Dificulta el trabajo colaborativo en versión gratuita
- ☐ Estructura y gestión de pruebas
- ☐ Integración con otras herramientas
- ☐ Mayor capacidad de automatización de pruebas. Groovy Script
- ☐ Posibilidad de integración en proyecto Java e importar librerías Java en herramienta SoapUI
- ☐ Curva de aprendizaje superior a Postman



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**

- ☐ Pruebas de servicios REST en Java
- ☐ Sin interfaz de usuario
- ☐ Integración con otras herramientas y proyectos Java
- ☐ Flexibilidad en la estructuración de la información (código)
- ☐ Automatización de pruebas en Java. Integración en tests E2E
- ☐ Orientado a perfiles DEV en exclusiva

Pruebas de Integración - Rest Assured



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**





# Pruebas E2E

- ❑ Se prueba el SUT desde el punto de vista del usuario final del mismo
  - Identificar errores en el interfaz de usuario
  - Identificar errores de comunicación entre sistemas
  - Identificar errores en la definición del flujo funcional

- ❑ Tratar el SW de Automatización como SW y no como un “conjunto de scripts”
  - Control de versiones del código fuente del SW de Automatización
  - Análisis estático del código fuente si es posible
  - Usar repositorios especializados para artefactos y resultados

#### ☐ Diseño del SW de Automatización

- Especificación Plan de Pruebas Regresión en base a riesgos y criticidad para el Negocio
- Estrategia orientada a procesos de negocio y no a casos de prueba (siempre que sea posible)
- Utilizar patrones de diseño (objeto página)

## Pruebas E2E

### ¿Qué buscamos en una herramienta de Automatización de Pruebas E2E?

- ☐ Desarrollo de pruebas rápido
- ☐ Permita la construcción de componentes reutilizables
- ☐ Facilite el trabajo colaborativo
- ☐ Multiplataforma, multidispositivo, multinavegador
- ☐ Integración con otras herramientas de nuestro ecosistema
- ☐ Generación de evidencias
- ☐ Permita paralelizar la ejecución de pruebas

# Herramientas de Automatización

Open Source



**cypress.io**



**ZAHORI**



The Appium logo, featuring a stylized 'A' made of two overlapping circles, one blue and one purple.  
**appium**  
automation for mobile apps

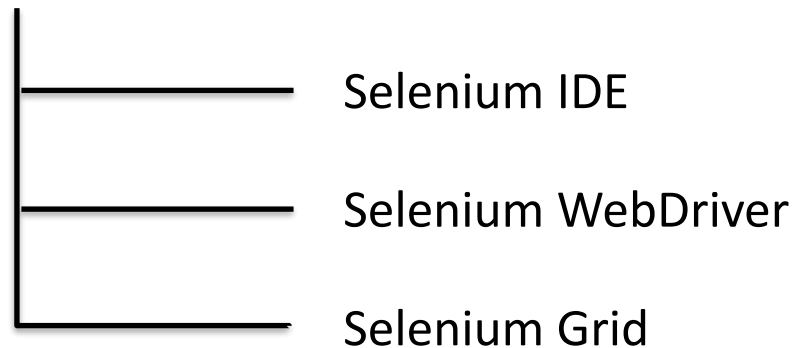
# Herramientas de Automatización

## Comerciales



### ¿Qué es Selenium?

- ❑ Selenium es una herramienta open source para automatización de pruebas en aplicaciones web, dando soporte a distintos navegadores y plataformas.
- ❑ Creado en 2004 por Jason Huggins
- ❑ Disponible en múltiples sabores y colores ;) ;)





### Titulares Selenium IDE

- ☐ Inicialmente conocido como Selenium Recorder
- ☐ Fuertemente vinculado al navegador Firefox al ser una extensión
- ☐ Permite grabar, editar y depurar pruebas
- ☐ Amplia comunidad involucrada en su mantenimiento y extensiones

### ¿Cuándo usamos Selenium IDE?

- ☐ “Ey! yo no soy programador”
- ☐ Adjuntar el script como evidencia de un bug
- ☐ Empezando a automatizar mi primera Test Suite
- ☐ Para automatizar tareas (web) repetitivas y aburridas

# Herramientas de Automatización

## Selenium IDE

The screenshot shows the Selenium IDE 1.9.1 interface. The window title is "Valid\_login.html - Selenium IDE 1.9.1". The menu bar includes File, Edit, Actions, Options, and Help. The Base URL bar shows "http://newtours.demoaut.com/". The toolbar includes buttons for Fast, Slow, and various test actions. The Test Case pane on the left lists "Invalid\_login" and "Valid\_login". The Table pane in the center displays a table of test steps. The Log/Reference/ UI-Element/ Rollup pane at the bottom shows the execution log, including an error message: "[error] Actual value 'Sign-on: Mercury Tours' did not match 'Find a Flight: Mercury Tours:'".

Menu bar

Base URL bar

Toolbar

Test Case pane

Log/Reference/ UI-Element/ Rollup pane

Command	Target	Value
open	/	
type	userName	tutorial
type	password	cause_of_failure
clickAndWait	login	
verifyTitle	Find a Flight: Mercu...	
clickAndWait	link=SIGN-OFF	
verifyTitle	Sign-on: Mercury T...	
clickAndWait	link=Home	
verifyTitle	Welcome: Mercury ...	

Runs: 2  
Failures: 1

Log/Reference/ UI-Element/ Rollup

[info] Changed test case  
[info] Executing: |open| / | |  
[info] Executing: |type| userName | tutorial |  
[info] Executing: |type| password | cause\_of\_failure |  
[info] Executing: |clickAndWait| login | |  
[info] Executing: |verifyTitle| Find a Flight: Mercury Tours: | |  
[error] Actual value 'Sign-on: Mercury Tours' did not match 'Find a Flight: Mercury Tours:'  
[info] Executing: |clickAndWait| link=SIGN-OFF | |  
[error] Element link=SIGN-OFF not found

### Setup Entorno. ¿Qué necesitamos?

- ☐ Navegador Firefox instalado
- ☐ Extensión Selenium IDE para Firefox  
<https://addons.mozilla.org/es/firefox/addon/selenium-ide/>



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**

### Titulares Selenium WebDriver

- ❑ Soporte en varios lenguajes de programación



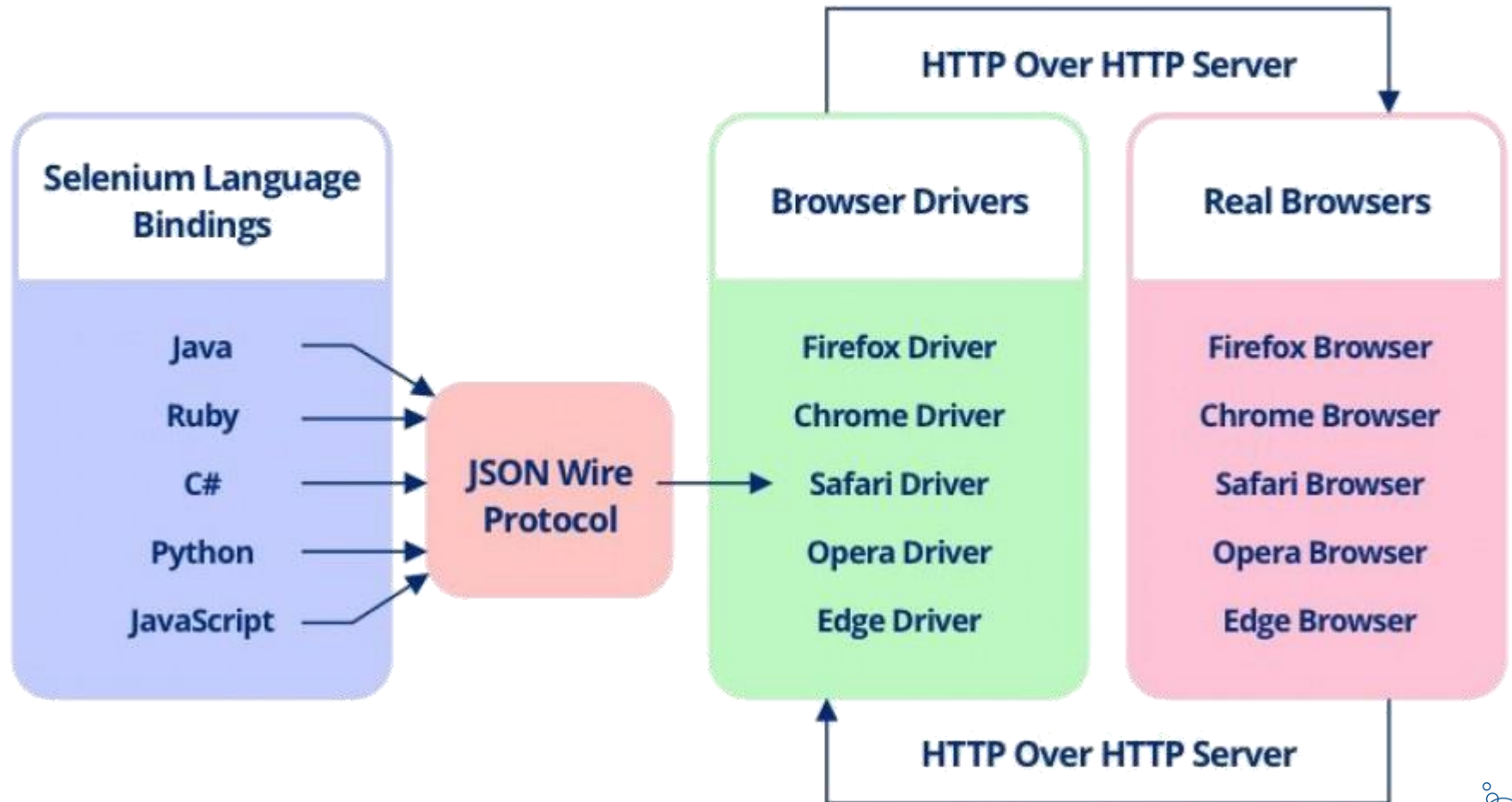
- ❑ Soporte múltiples navegadores



- ❑ Soporte múltiples plataformas



# Selenium WebDriver Architecture



### Clases librería cliente Java

Nombre de la clase	Descripción
WebDriver	Representa el navegador web
Options	Utilizado para gestionar opciones del driver
Timeouts	Utilizado para setear timeouts relativos a esperas implícitas, tiempos de carga de página, etc
Window	Utilizado para gestionar la ventana del navegador
WebElement	Representa el elemento web
By	Representa la información necesaria para localizar elementos web
WebDriverWait	Utilizado para realizar esperas de eventos en el navegador
Actions	Utilizado para realizar acciones “complejas”



### Localización elementos web (clase By)

Tipo
className
cssSelector
id
linkText
name
partialLinkText
tagName
xpath



### Setup Entorno. ¿Qué necesitamos?

- ☐ Eclipse IDE
- ☐ Java SDK (ejemplo Amazon Corretto 8)  
<https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/downloads-list.html>
- ☐ Maven  
<https://maven.apache.org/download.cgi>
- ☐ Navegador Google Chrome



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**


# Herramientas de Automatización

## Selenium Grid

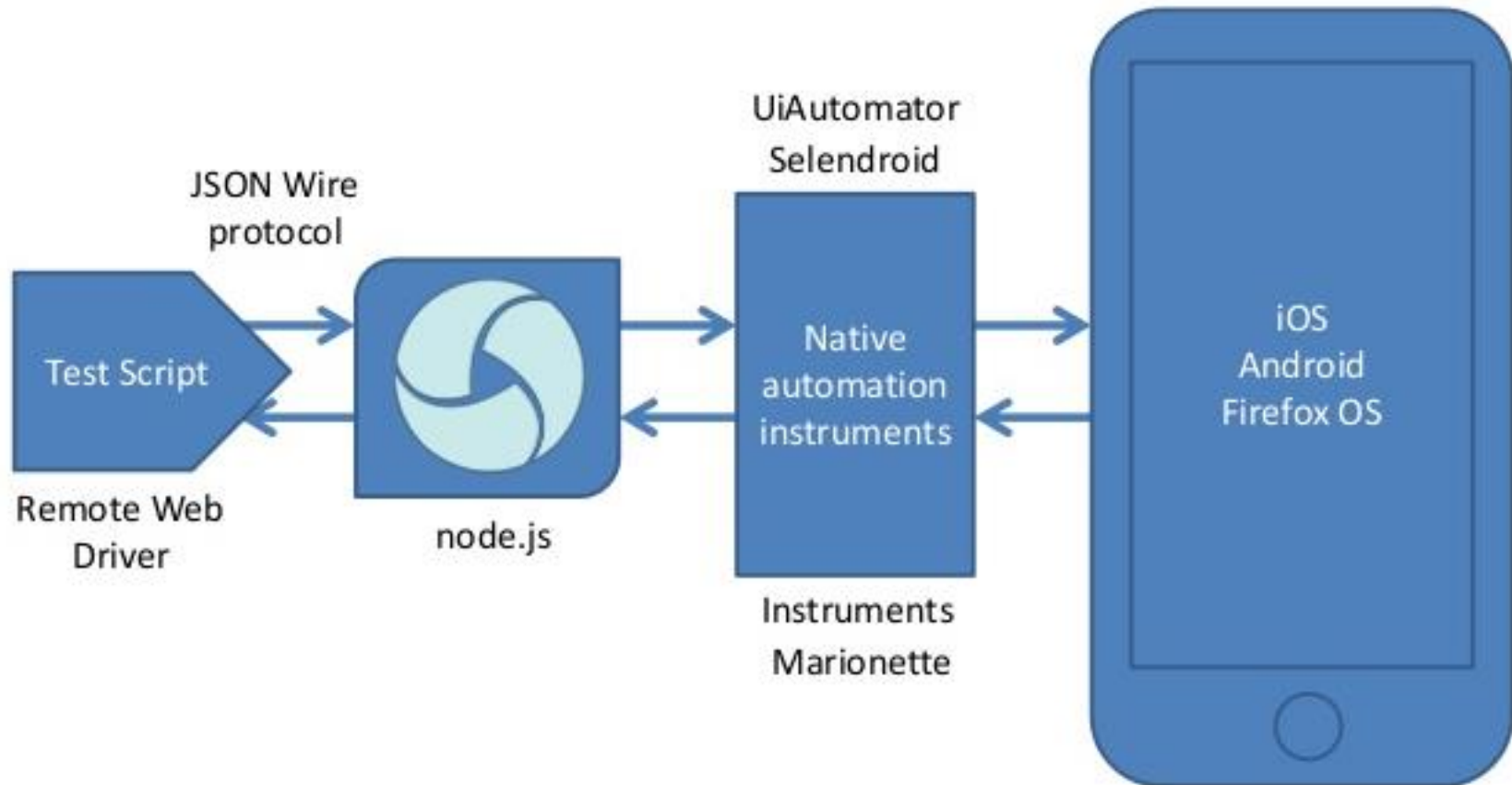
Clases libería cliente
Capabilities
RemoteWebDriver



### Titulares Appium

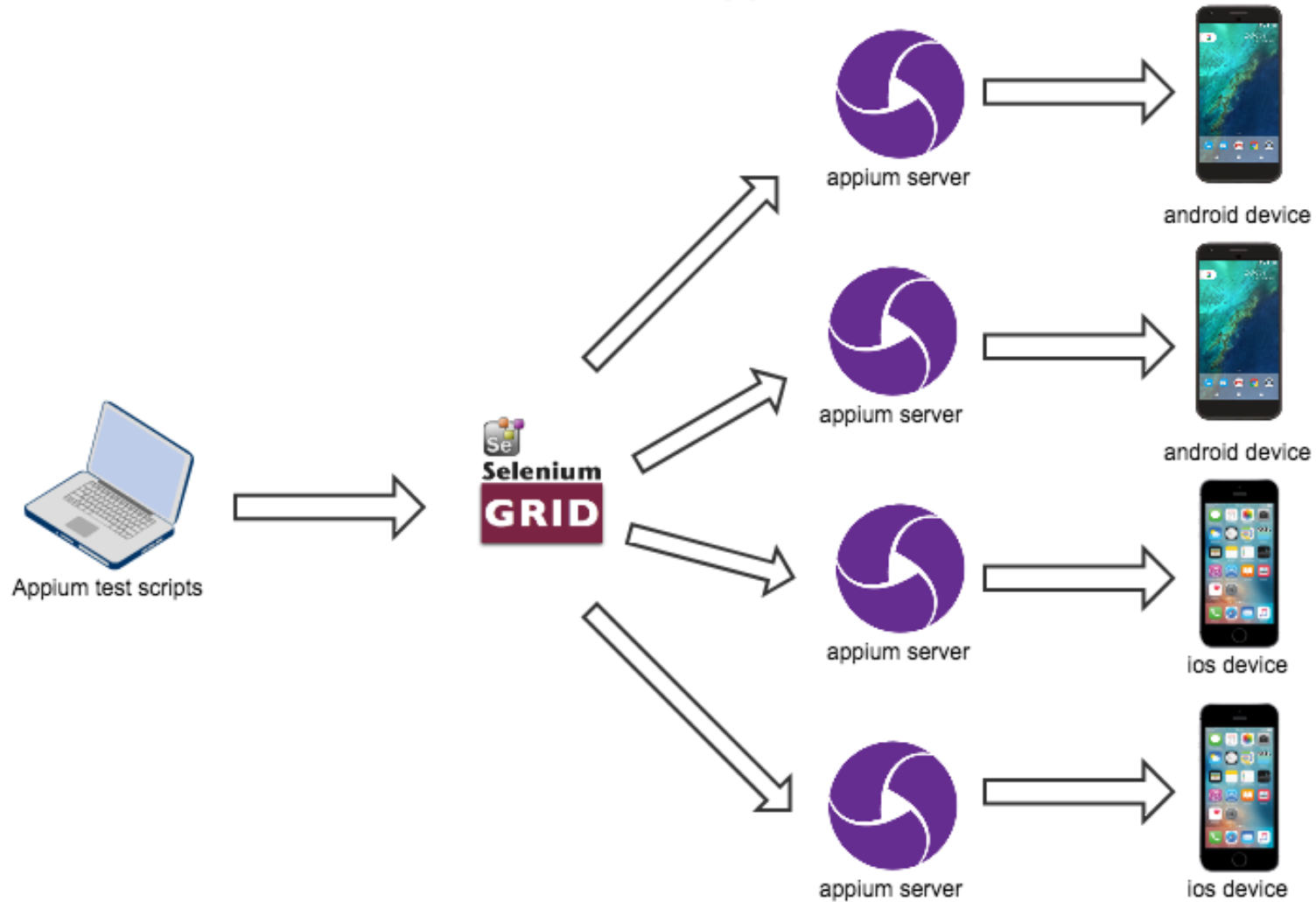
- ☐ Es una herramienta OpenSource de automatización de pruebas multiplataforma para aplicaciones móviles
- ☐ Soporta automatización APPs nativas, híbridas y web mobile
- ☐ Basado en JSON Wire protocol y servidor Appium escrito en NodeJS
-  Coste de la automatización de pruebas en APPs

## Appium



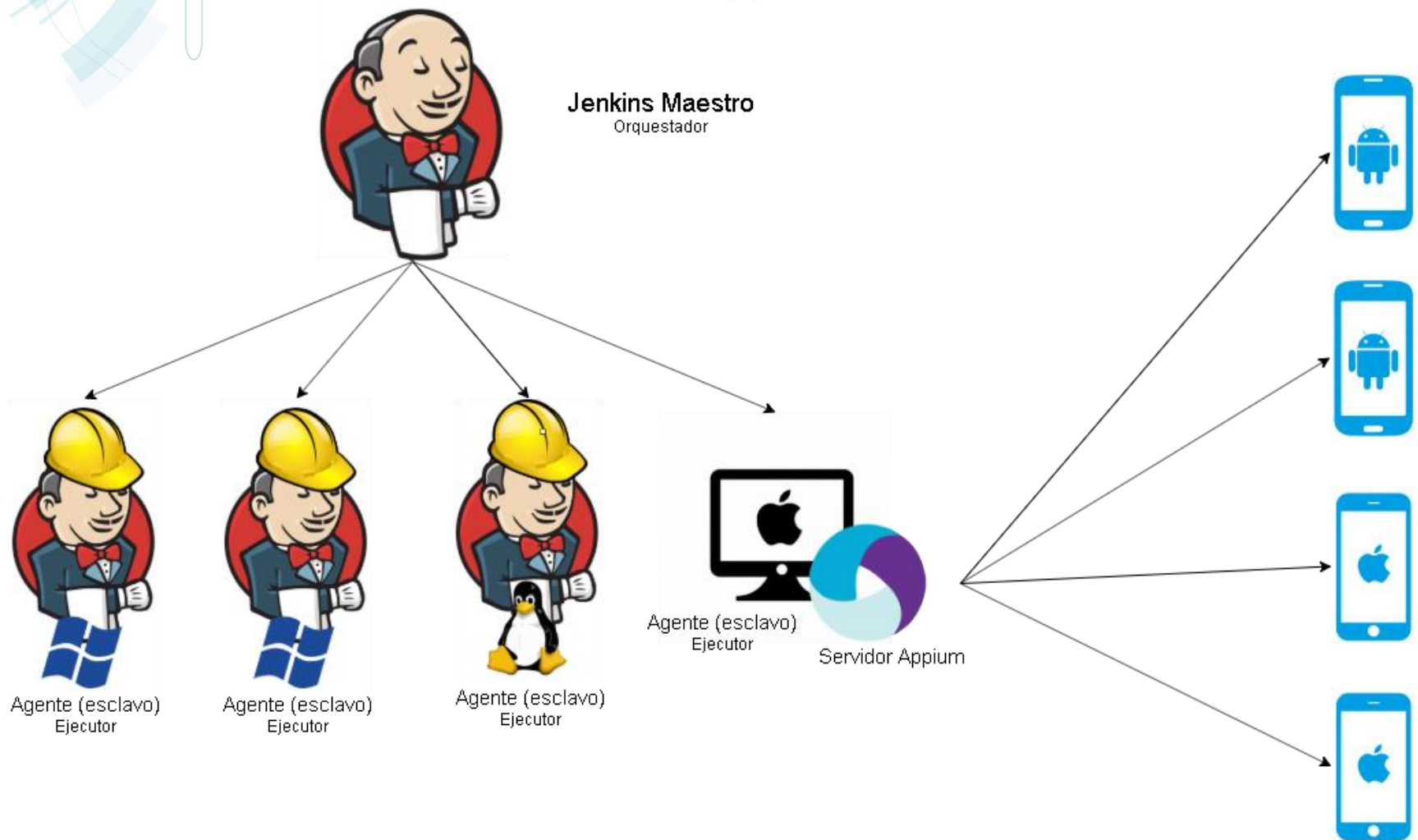
# Herramientas de Automatización

## Appium



# Herramientas de Automatización

## Appium





# Herramientas de Automatización

## Appium



### ¿Qué es Cypress?

- ☐ Cypress es un framework de pruebas para pruebas E2E automáticas que no utiliza Selenium. Los comandos cypress se ejecutan en el mismo ciclo de ejecución que la aplicación web.
- ☐ Creado en 2014
- ☐ Open source + versión pago
  - Cypress Dashboard
- ☐ Developer-friendly (only)

### Características Cypress

- ☐ Viajes en el tiempo
- ☐ Reejecución automática con cambios en la aplicación / fuentes de tests E2E
- ☐ Intercepción de peticiones HTTP del browser
- ☐ Generación automática de evidencias (capturas de pantalla y vídeo)
- ☐ Velocidad de ejecución más rápida que Selenium
- ☐ Infinidad de plugins para extender sus capacidades por defecto

### Características Cypress

- ❑ Lenguajes de programación



- ❑ Soporte múltiples navegadores



- ❑ Soporte múltiples plataformas



Setup Entorno. ¿Qué necesitamos?



Cypress



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**



# Pruebas de Rendimiento

- ❑ Someten al SUT a una carga de trabajo con el fin de medir su velocidad, fiabilidad y estabilidad en esas condiciones de trabajo
  - ➔ Localizar cuellos de botella. Requiere monitorizar HW máquina/s
  - ➔ Localizar problemas de rendimiento de la aplicación (SW)
  - ➔ Verificar cumplimiento de SLAs



## Frontend



## Backend

- ☐ Métricas navegador
  - Tiempo renderizado
  - Tiempo interactivo
  - Tiempo carga

- ☐ Herramientas automáticas

- ☐ Respuesta del servidor

- ☐ Herramientas diseño / automatización de tests de rendimiento

### ❑ Visión tradicional escenarios Tests Rendimiento

## Estabilidad

El objetivo es medir con una carga soportada por el sistema (carga esperada) que el mismo no se degrada según avanza el tiempo

- Duración de la ejecución prolongada (varias horas)
- Número de usuarios concurrentes acorde a la carga esperada en un uso normal del SUT
- Se intercalan esperas entre peticiones para simular usuarios comunes del SUT

### ☐ Visión tradicional escenarios Tests Rendimiento

## **Rendimiento**

El objetivo es medir el número de peticiones por segundo que puede atender el sistema en condiciones de carga normal

- Duración de la ejecución ventana razonable para simular suficientes eventos en el SUT (~ 1h)
- Número de usuarios concurrentes levemente inferior a la carga esperada en un uso normal del SUT
- No se intercalan esperas entre peticiones

## ❑ Visión tradicional escenarios Tests Rendimiento

### Saturación

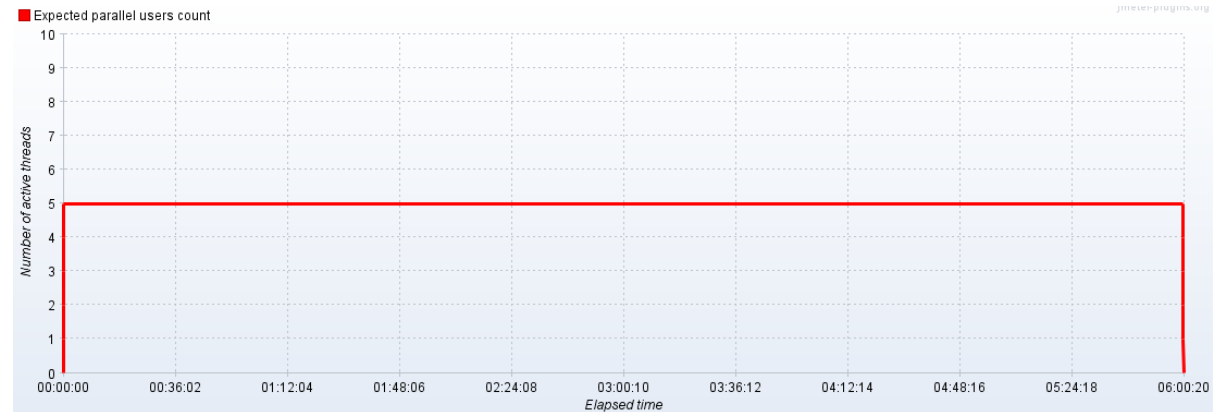
Se tiene 2 objetivos principales:

- Medir el número máximo de usuarios que el sistema soporta hasta que empieza a fallar debido a condiciones extremas.
- Comprobar si una vez que se ha provocado el fallo del sistema, en algún momento el mismo se recupera según se va descendiendo la carga.

- ➡ Se configura la ejecución en ramp-up y ramp-down
- ➡ La duración viene determinada por el número de escalones / etapas y la duración de los mismos
- ➡ El número de usuarios concurrentes debe alcanzar un pico suficiente para saturar el SUT
- ➡ No se introducen esperas entre peticiones

## ❑ Visión tradicional escenarios Tests Rendimiento

**Estabilidad  
Rendimiento**



**Saturación**

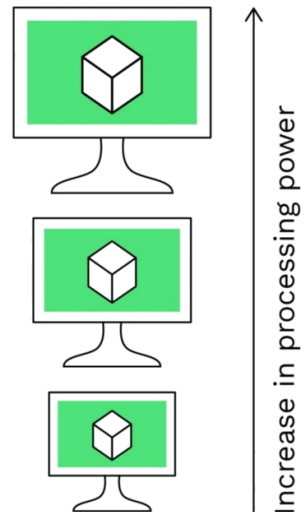


## ❑ Visión actual escenarios Tests Rendimiento

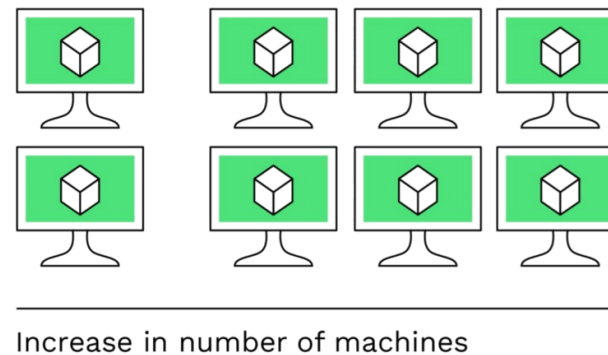
### ¿Qué ha cambiado?

#### Scalability

##### Vertical scaling



##### Horizontal scaling



❑ Visión actual escenarios Tests Rendimiento

### **¿Qué hacemos?**

- Analizar cada caso y aplicar soluciones a medida con la base de la visión tradicional
- Aplicar sentido común

## Pruebas de Rendimiento





- ☐ Open Source
- ☐ Interfaz de usuario obsoleto y poco intuitivo
- ☐ Multiplataforma
- ☐ Flexible, potente, extensible mediante plugins
- ☐ Permite ejecución distribuida en distintos nodos
- ☐ Permite grabar actividad de usuario mediante proxy intermedio
- ☐ Curva de aprendizaje elevada
- ☐ Generación de reportes HTML útiles para analizar resultados



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**



- ☐ Open Source + versión pago
- ☐ Herramienta CLI con APIs
- ☐ Multiplataforma
- ☐ Scripting en JS. Fácil implementación para devs ( ~~node.js~~ )
- ☐ No permite ejecución distribuida en distintos nodos (no necesario)
- ☐ Permite grabar actividad de usuario mediante extensión para Chrome y Firefox
- ☐ Curva de aprendizaje más reducida que JMeter
- ☐ Plataforma cloud (ejecución remota y reportes) de pago



**KEEP  
CALM**

**AND**

**CONSIDER  
AN EXAMPLE**



**Más información en:**

**[www.panel.es](http://www.panel.es)  
[panel@panel.es](mailto:panel@panel.es)**

**panel.es**  
**Panel Sistemas Informáticos, S.L.**  
**Consultoría, servicios y soluciones TI.**

