

SELENIUM

ÍNDICE

Tema 1: Introducción a tests funcionales. Enfoque de testing. Diseño de casos de prueba

Tema 2: Introducción a la automatización de tests. Conceptos básicos

Tema 3: Xenu. testing de links en página web

Tema 4: Selenium IDE. Comandos básicos

Tema 5: JUnit. Conceptos básicos

Tema 6: Selenium RC. Importación desde Selenium. IDE de edición de script Java en Eclipse

Tema 7: Ejecución de script en Selenium. Core desde Eclipse

ÍNDICE

Tema 8: Watir. Ejecucion de comandos básicos

Tema 9: JMeter (Básico) elementos basicos de JMeter como herramienta de test funcional grabación/ejecución de scripts

Tema 10: JMeter (Intermedio). Explicación de elementos avanzados para test casos complejos. Producción de reportes y evidencias. Ejecución de scripts JMeter desde CLI.

Tema 11: Soap UI. Elementos básicos. Test de web services. Generación de mock

Tema 12: Mockito. Conceptos básicos de mock para JUnit

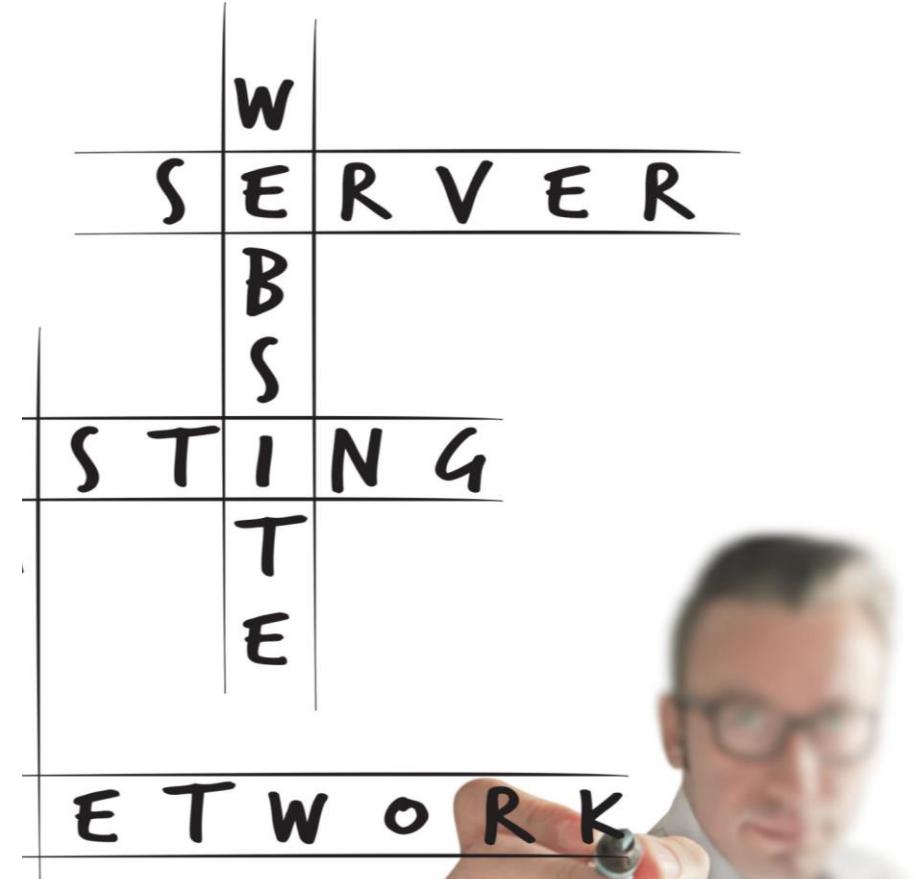
Tema 13: BadBoy. Introducción.

PRESENTACIÓN

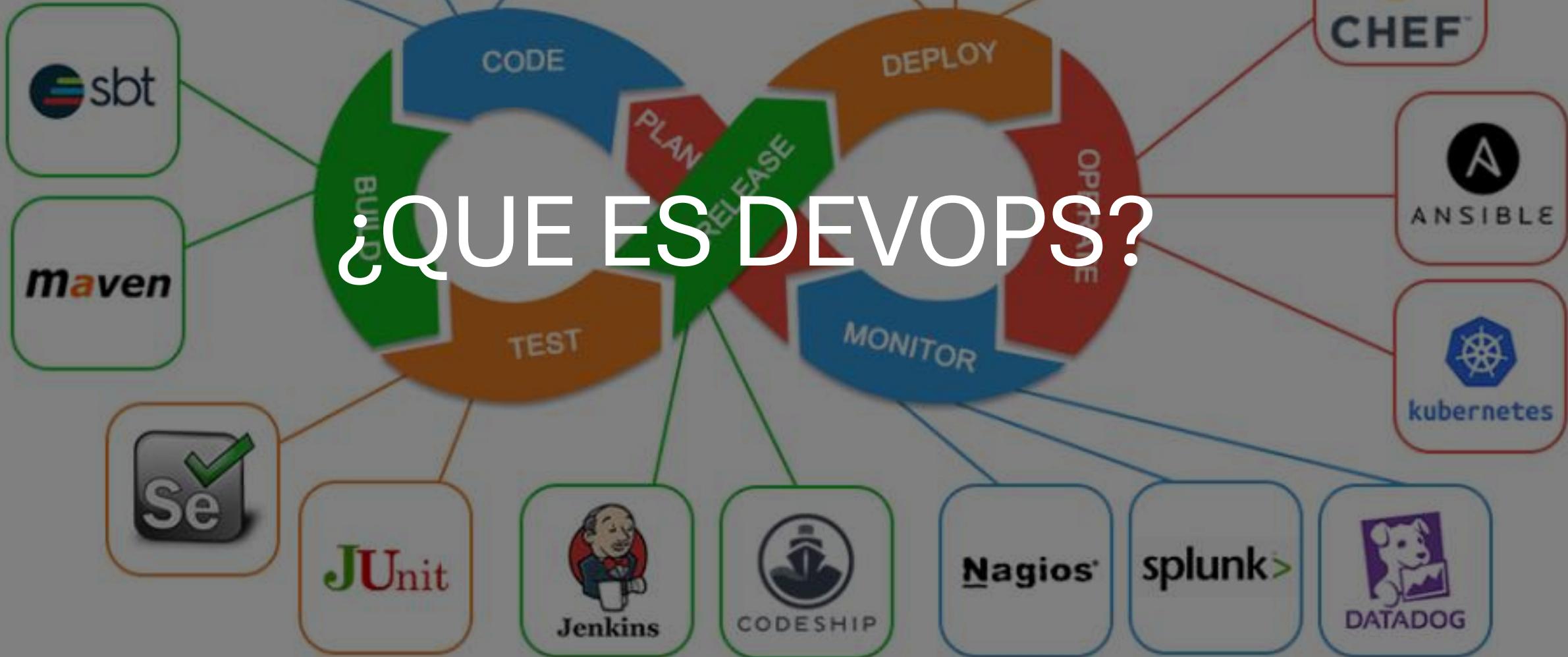


Introducción a Tests Funcionales

- Los tests funcionales validan que el software cumple con los requisitos del usuario.
- Enfocan en cómo el sistema debe comportarse bajo condiciones específicas.
- El diseño efectivo de casos de prueba es crucial para garantizar la calidad.



¿QUE ES DEVOPS?



VENTAJAS



Una mejor y más rápida entrega de productos



Resolución de problemas en menos



Mejor escalabilidad y disponibilidad



Entornos de funcionamiento más estables



Mejor utilización de los recursos



Mayor automatización



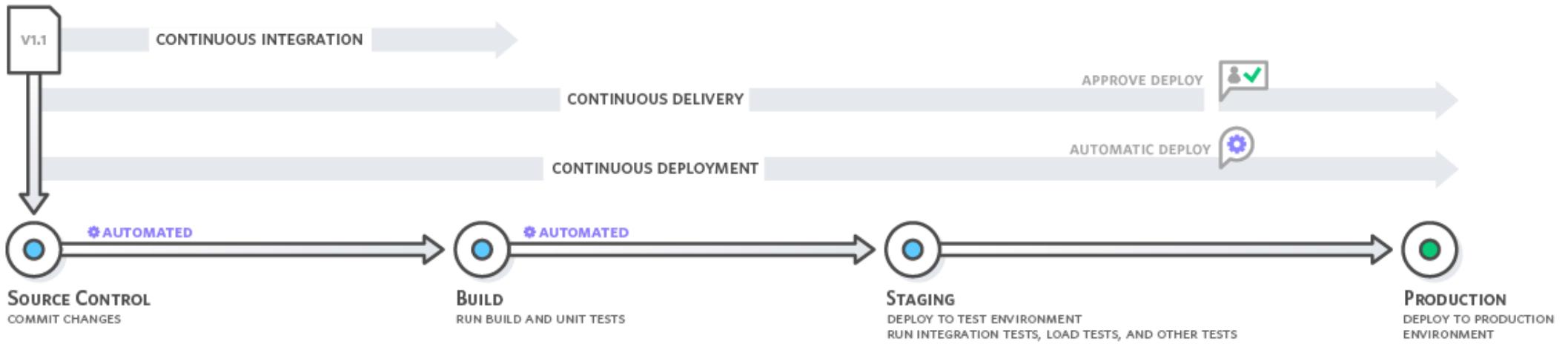
Mayor visibilidad de resultados del sistema



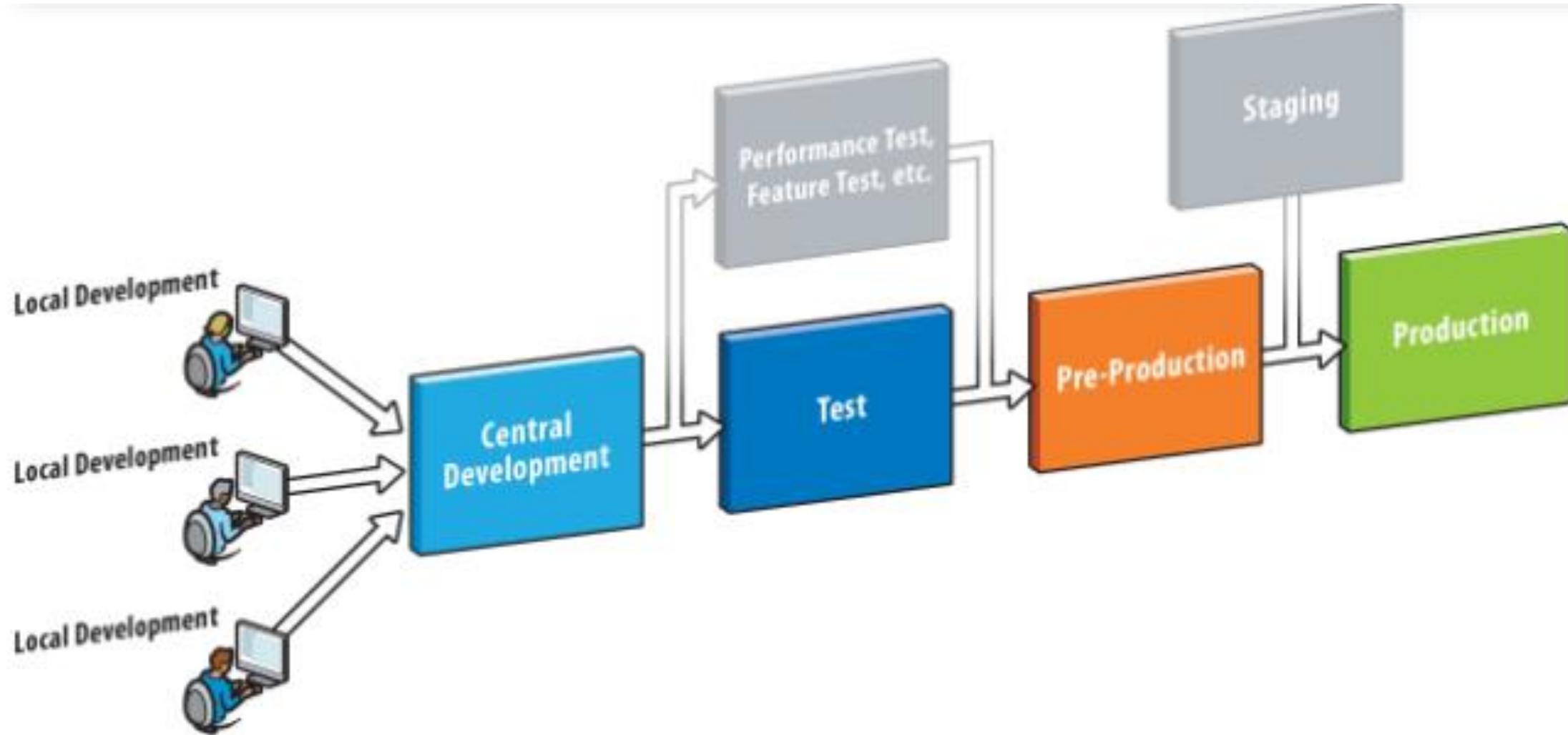
Mayor innovación



INTEGRACIÓN CONTINUA



ENTORNOS



ENTREGA CONTINUA



CALIDAD DE SOFTWARE

0

Bugs

207

Vulnerabilities

0

New Bugs

207

New Vulnerabilities

Code Smells

155d

Distr

4.4k

Code Smells

154d

New Distr

4.3k

New Code Smells

Coverage



66.7%

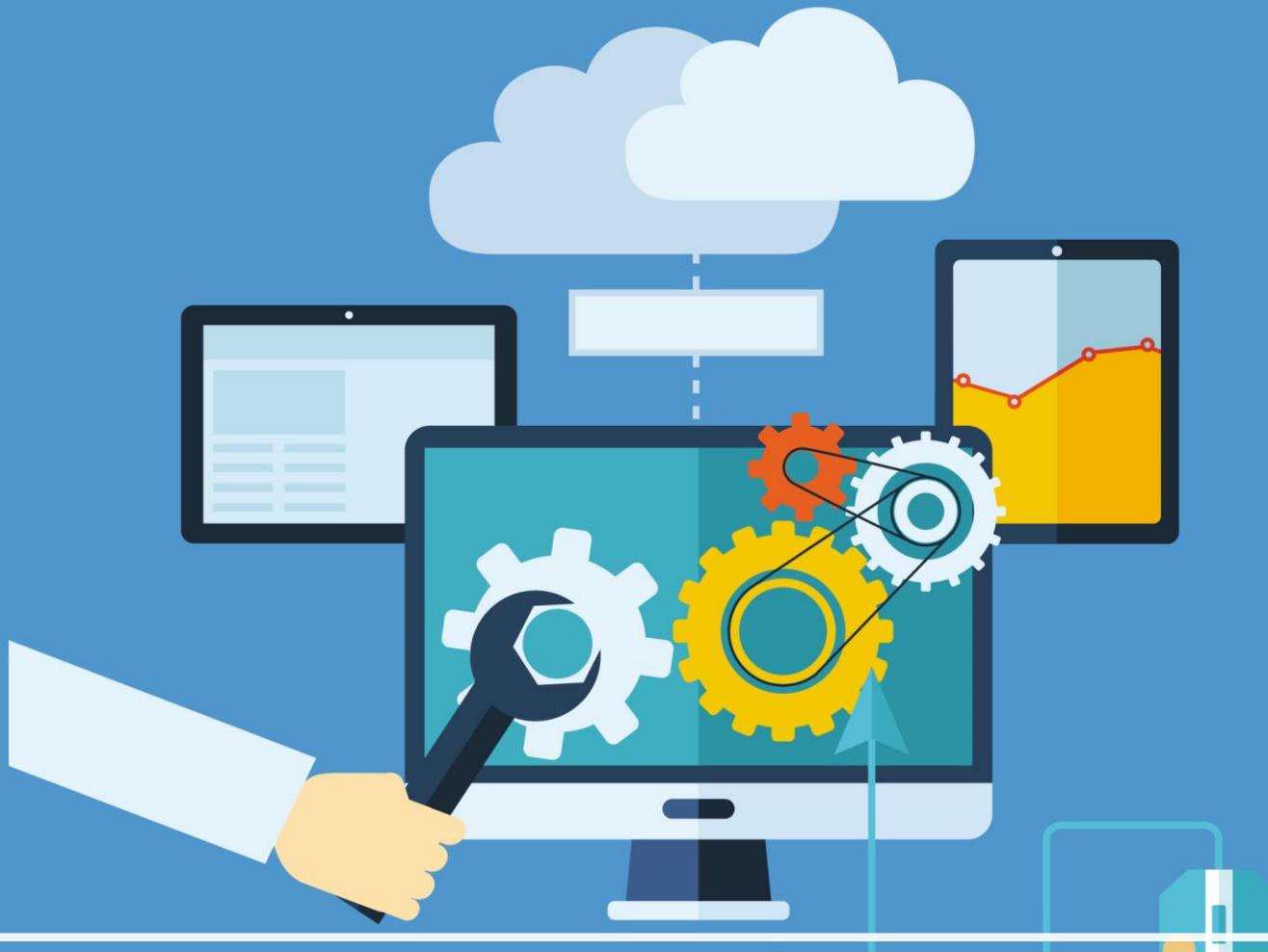
Coverage

14k

Unit Tests

Coverage on New Code

Duplications



INTRODUCCIÓN A PRUEBAS



vs.



Pruebas
dinámicas

Pruebas
estáticas

TIPOS DE SOFTWARE TESTING

Pruebas funcionales

- Pruebas unitarias
- Pruebas de integración
- Pruebas de sistema
- Pruebas de sanidad
- Pruebas de humo
- Pruebas de interfaz
- Pruebas de regresión
- Pruebas de aceptación

Pruebas no funcionales

- Pruebas de rendimiento
- Prueba de carga
- Pruebas de estrés
- Pruebas de volumen
- Pruebas de seguridad
- Pruebas de compatibilidad
- Pruebas de instalación
- Pruebas de recuperación
- Pruebas de confiabilidad
- Pruebas de usabilidad
- Pruebas de conformidad
- Pruebas de localización



Test de Caja Negra



Test de Caja Blanca

Introducción a la Automatización de Tests

- La automatización de tests mejora la eficiencia y la cobertura de pruebas.
- Permite ejecutar pruebas repetitivas de manera rápida y consistente.
- Es fundamental seleccionar las herramientas adecuadas para cada tipo de prueba.



Estrategia de Pruebas

- Planificación: Definición de estrategias, alcance, recursos y calendario de pruebas.
- Diseño de Casos de Prueba: Creación de escenarios y casos de prueba basados en requisitos.
- Ejecución de Pruebas: Ejecución de casos de prueba y registro de resultados.
- Análisis de Resultados: Evaluación de los resultados de las pruebas y seguimiento de los defectos encontrados.
- Informe y Retest: Creación de informes de pruebas y reevaluación tras la corrección de defectos.



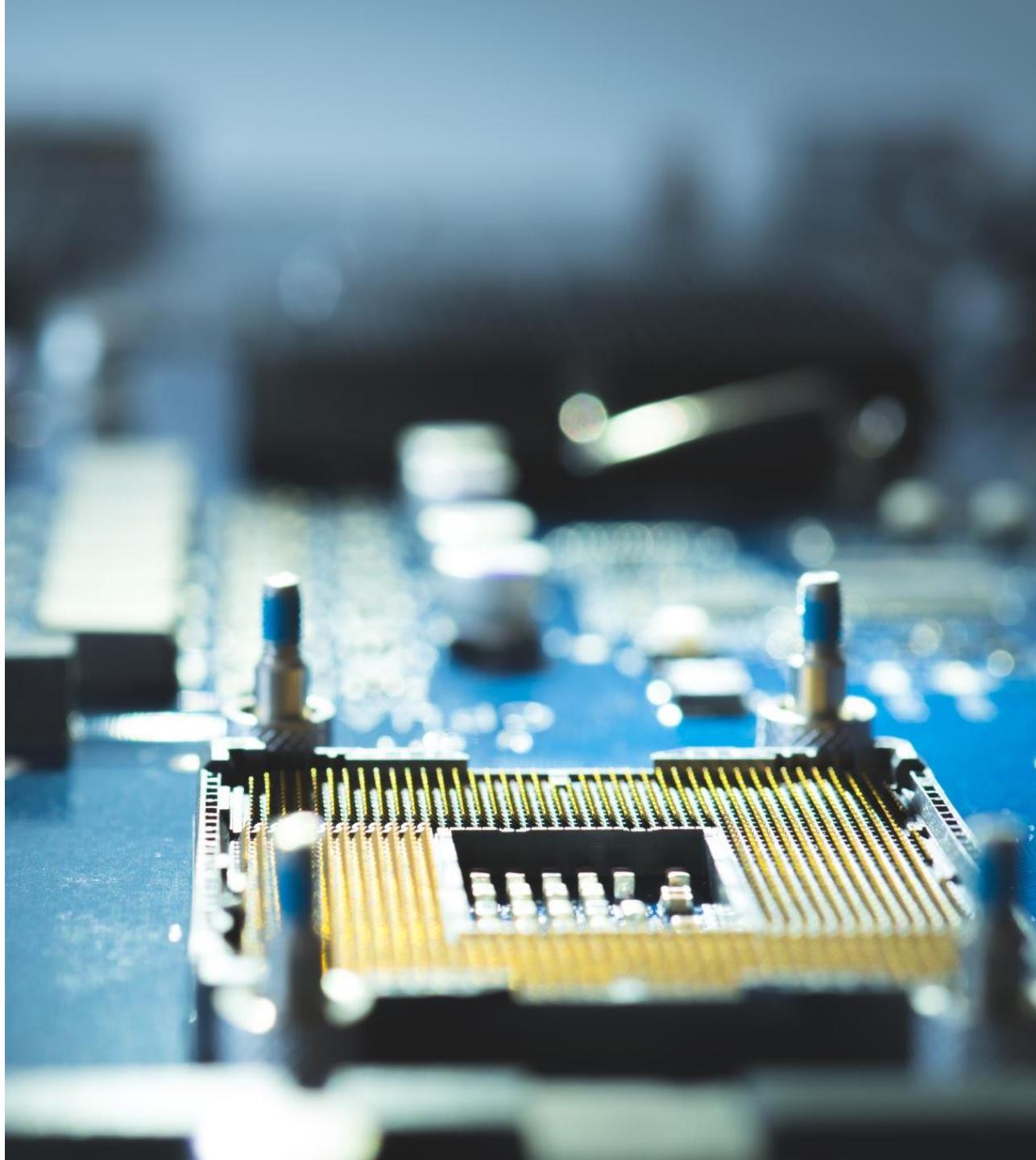
Diseño de Pruebas

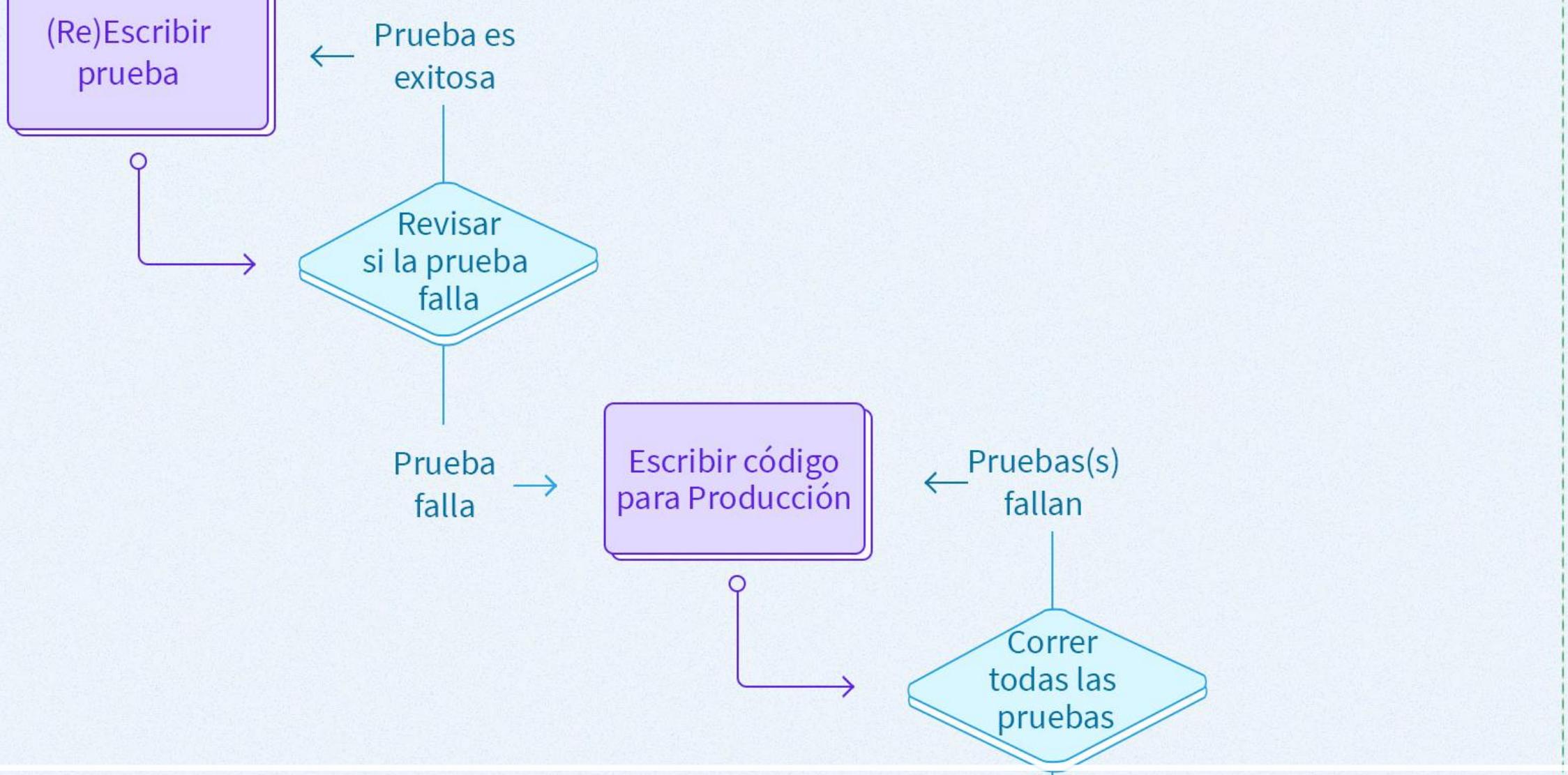
- Conocimiento de los Requisitos
- Reducción de la funcionalidad a probar
- Definición de Casos de Prueba
- Relacionar con los Casos de Usuario (historias)
- Pensar bien los costes de las pruebas
- Enfocarse en las funcionalidades más críticas
- Cada prueba requiere su herramienta



HERRAMIENTAS

- Pruebas Unitarias: Junit
- Pruebas de Integración: Mockito
- Pruebas de Aceptación/Funcionalidad Web: Selenium
- Bases de Datos/Web Rendimiento: Jmeter/Blazemeter/Gatling
- Servidor de Integración Contínua: Jenkins/TravisCI/CircleCI
- Herramienta de Seguimiento de Proyectos e Incidencias: Jira/Mantis
- Automatizador de la Compilación, Despliegue y Ejecución de proyectos: Maven/Gradle/NPM
- Despliegue de aplicaciones: Docker/Kubernetes





AGILE TESTING

Todas las pruebas
son exitosas →

Limpiar el
código



AUTOMATIZACIÓN DE LAS PRUEBAS

Xenu: Testing de Links en Página Web

- Xenu es una herramienta para verificar enlaces rotos en sitios web.
- Proporciona un informe detallado de enlaces que no funcionan.
- Es útil para mantener la integridad de los enlaces en el contenido web.

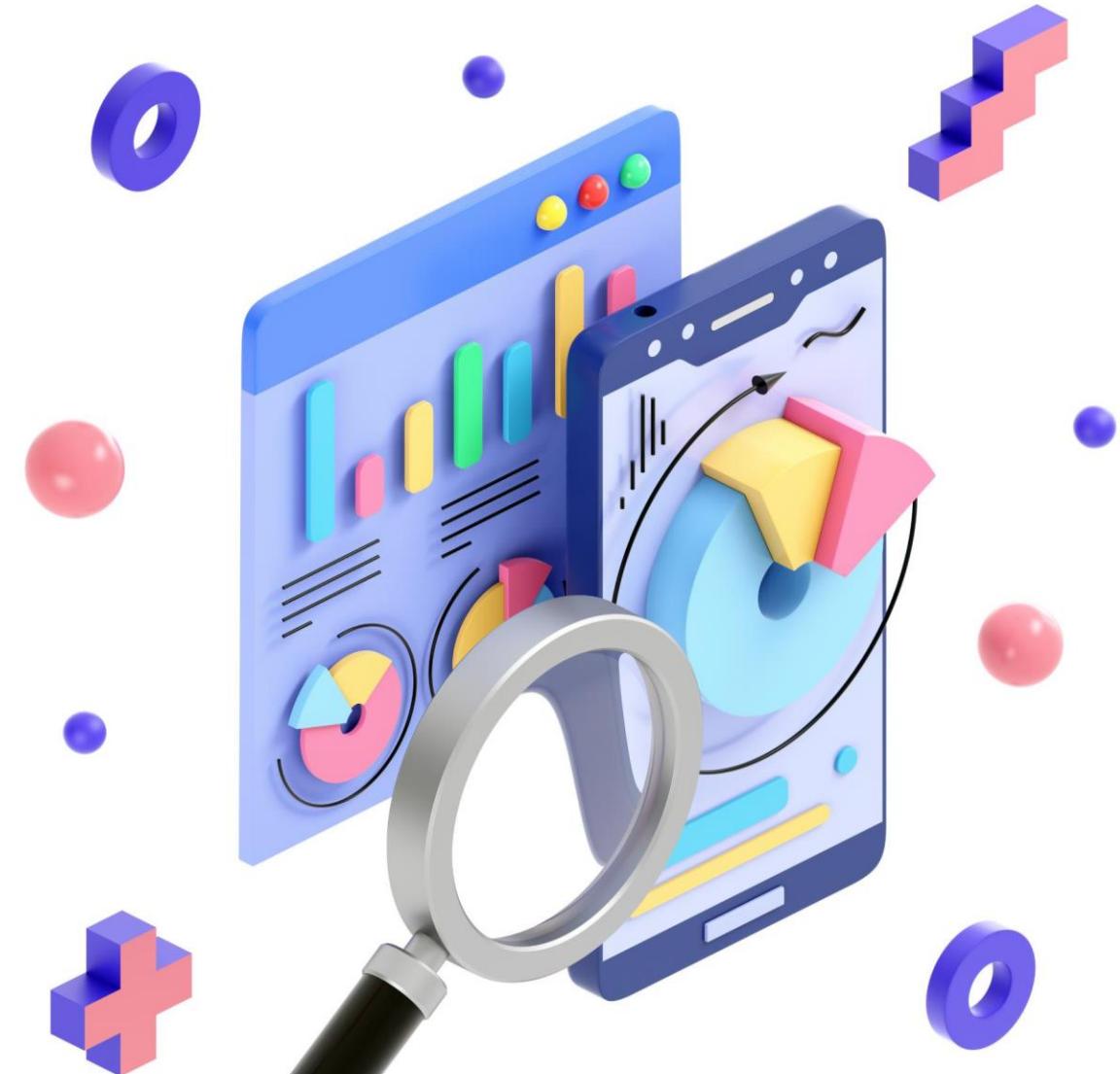
Desarrollador y contexto de la creación

Desarrollo de la Herramienta

Xenu's Link Sleuth fue creado para satisfacer la necesidad de herramientas efectivas en el análisis de enlaces web, mejorando la eficiencia de los administradores de sitios.

Uso en SEO

Desde su lanzamiento, Xenu's Link Sleuth ha sido adoptado por profesionales del SEO para optimizar la salud y rendimiento de los sitios web.





Compatibilidad y entorno de funcionamiento

Compatibilidad del Sistema Operativo

La herramienta es compatible con sistemas operativos Windows, asegurando un amplio acceso para los usuarios.

Diversos Entornos de Trabajo

Funciona en diversos entornos, lo que permite su uso en diferentes configuraciones de trabajo y necesidades.

Diseño Simple y Accesible

Su diseño simple permite a usuarios de diferentes niveles aprovechar sus funciones sin complicaciones.

Comparación con alternativas comerciales

Herramientas Comerciales

Existen múltiples herramientas comerciales disponibles que ofrecen diversas funciones para los usuarios con diferentes necesidades y presupuestos.

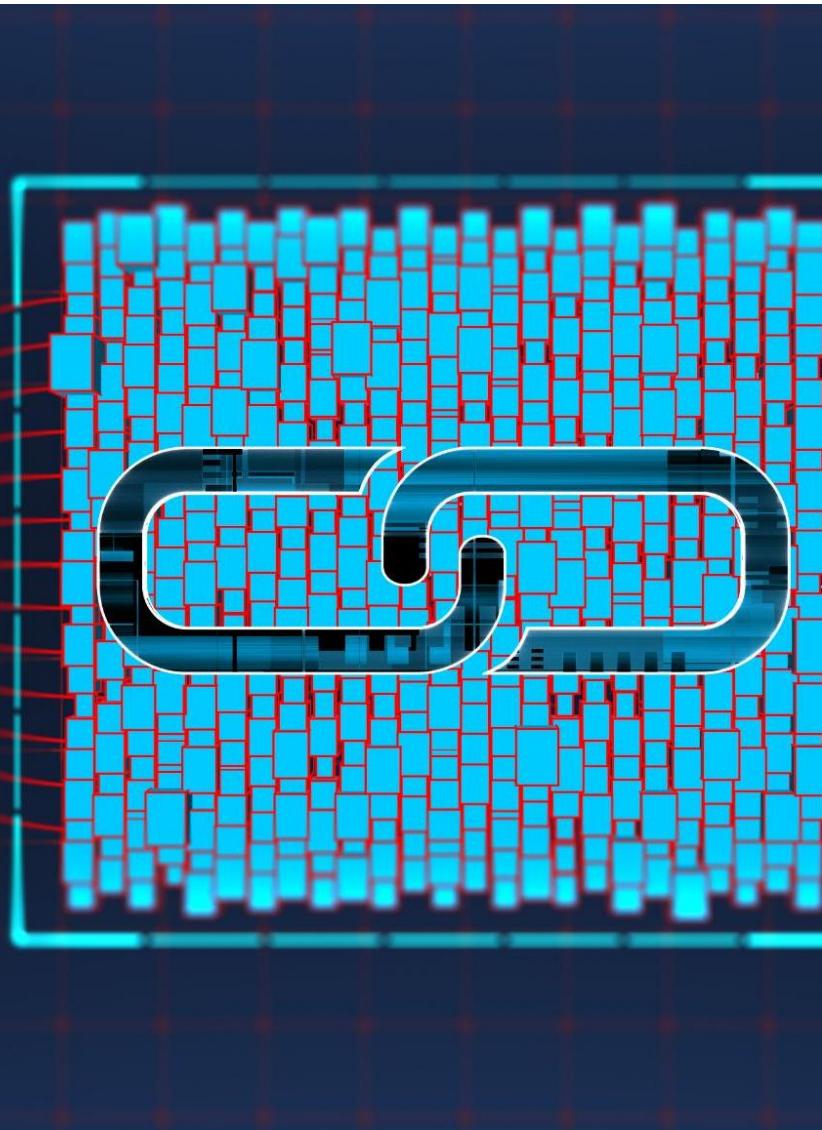
Ventajas de Xenu

Xenu se destaca por ser una herramienta gratuita y efectiva, permitiendo a los usuarios tener acceso a funcionalidades sin costo.

Facilidad de Uso

La facilidad de uso de Xenu es una de sus principales ventajas, lo que la hace accesible para usuarios con diferentes niveles de experiencia.





Proceso de rastreo y recopilación de datos

Eficiencia del rastreo

El proceso de rastreo de Xenu es altamente eficiente, permitiendo un análisis rápido de sitios web.

Identificación de problemas

La herramienta ayuda a identificar enlaces rotos y redirecciones, facilitando la mejora del sitio web.

Oportunidades de mejora

Al recopilar datos, los usuarios pueden encontrar oportunidades para optimizar su sitio web y su rendimiento.



Interfaz de usuario y experiencia de uso

Interfaz Clara y Sencilla

La interfaz de Xenu es intuitiva, permitiendo a los usuarios navegar sin complicaciones, facilitando el acceso a las funciones.

Opciones de Menú Organizadas

Las opciones de menú están bien organizadas para que los usuarios puedan encontrar rápidamente lo que necesitan, mejorando la eficiencia.

Optimización para Todos los Usuarios

El diseño de la interfaz está optimizado tanto para principiantes como para expertos, garantizando una experiencia de uso satisfactoria.

Tipos de información que registra

Enlaces Rotos

Xenu identifica enlaces rotos que pueden afectar la experiencia del usuario y el SEO del sitio. Corregir estos enlaces es vital para mantener la autoridad del dominio.

Errores de Servidor

Los errores de servidor son problemas que pueden impedir que los usuarios accedan a ciertas páginas. Identificarlos ayuda a mejorar la estabilidad del sitio web.

Redirecciones

Xenu también registra redirecciones que pueden afectar la jerarquía de URL y la experiencia del usuario. Optimizar estas redirecciones es crucial.





Definición y funciones de un crawler

Qué es un Crawler

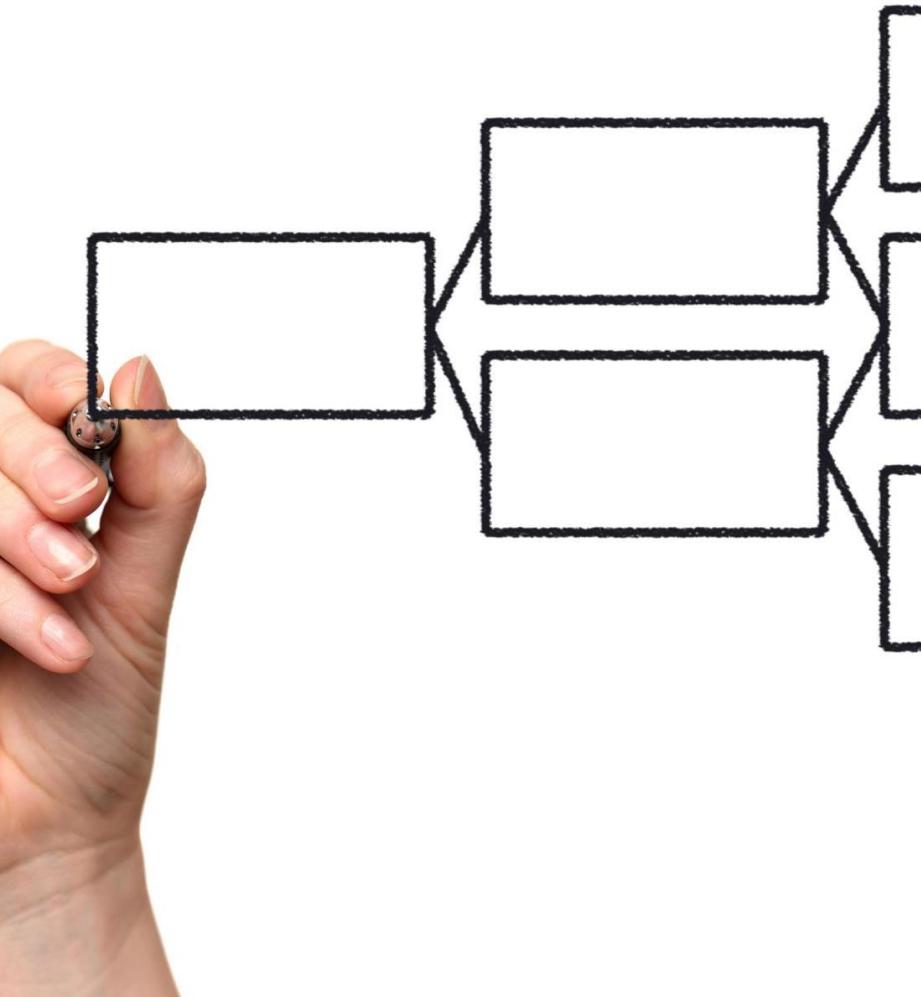
Un crawler, o araña web, es un programa automatizado que explora internet para recolectar datos de sitios web.

Función de Recopilación

Los crawlers recopilan información sobre el contenido de los sitios web, que es crucial para la indexación por parte de los motores de búsqueda.

Optimización para SEO

Optimizar los crawlers es vital para mejorar la visibilidad y el posicionamiento en los resultados de búsqueda, lo que impacta el tráfico web.



Ejemplos de uso en auditorías y análisis web

Identificación de Problemas de SEO

Xenu ayuda a detectar problemas de SEO, asegurando que el sitio web está optimizado para los motores de búsqueda.

Detección de Enlaces Rotos

Un uso clave de Xenu es la detección de enlaces rotos, lo que mejora la experiencia del usuario y la salud del sitio.

Evaluación de la Estructura del Sitio

Evaluar la estructura del sitio es crucial para el SEO, y Xenu proporciona informes detallados para mejorar la accesibilidad.

Capacidades de rastreo y análisis

Rastreo eficiente de sitios

Xenu permite rastrear grandes sitios web rápidamente, lo que facilita auditorías exhaustivas y mejora la eficiencia de análisis.

Análisis detallado

Las capacidades de análisis de Xenu proporcionan información valiosa sobre la estructura del sitio y posibles áreas de mejora.





Opciones de reporte y visualización de datos

Diversas opciones de reporte

La herramienta presenta múltiples formatos de reporte, adaptándose a las necesidades de los usuarios para facilitar la comprensión de los datos.

Visualización clara de datos

Los datos se visualizan de manera clara y comprensible, lo que ayuda a los usuarios a entender la información rápidamente.

Toma de decisiones informadas

La visualización efectiva de datos facilita la toma de decisiones informadas, mejorando así el rendimiento del sitio web.



Costo y accesibilidad

Herramienta Gratuita

Xenu se destaca por ser una herramienta gratuita, eliminando barreras de costo para los usuarios.

Accesibilidad para Todos

La accesibilidad de Xenu permite su uso tanto por pequeños propietarios como por grandes empresas, promoviendo igualdad en el acceso a recursos.

Fomento de la Comunidad SEO

Esta herramienta gratuita ha fomentado una mayor colaboración y uso entre la comunidad SEO, mejorando las prácticas de optimización.

DEMO



Ventajas competitivas de Xenu's Link Sleuth

Simplicidad de Uso

Xenu se destaca por su interfaz intuitiva, lo que facilita su uso incluso para principiantes en SEO.

Manejo de Sitios Grandes

La herramienta es altamente eficiente en la gestión de grandes sitios web, lo que la hace ideal para desarrolladores y SEO.

Gratuidad

Xenu es completamente gratuita, lo que la convierte en una opción atractiva en comparación con otras herramientas de pago.





Limitaciones y desventajas

Falta de Funciones Avanzadas

Xenu carece de varias funciones avanzadas que están disponibles en herramientas comerciales, limitando su capacidad para usuarios exigentes.

Análisis Menos Profundos

Los usuarios que buscan análisis más profundos pueden encontrar que las limitaciones de Xenu no satisfacen sus necesidades específicas.





LABORATORIO

JUnit: Conceptos Básicos

- JUnit es un marco de pruebas para aplicaciones Java.
- Permite crear y ejecutar tests unitarios de manera estructurada.
- Es fundamental en el desarrollo ágil y en la metodología TDD.





EJEMPLO

```
import org.junit.Test;  
import static org.junit.Assert.assertEquals;  
public class MiClaseTest {  
  
    @Test  
    public void pruebaSuma() {  
        int resultado = MiClase.suma(3, 5);  
        assertEquals(8, resultado);  
    }  
}
```



MENSAJES PERSONALIZADOS

```
assertEquals(2, calculadora.suma(1, 1),  
“La suma debería ser 2”);  
  
assertEquals(2, calculadora.suma(1, 1), () -  
> “La suma debería ser 2”)
```

LA ANATOMIA DE JUNIT

```
package examples.nbank;

public class Conversion {

    public double tempConversion (double temperature, String unit) {
        if (unit.equals("F"))
            return (temperature - 32) * (5.0/9.0);
        else
            return (temperature * (9.0/5.0)) + 32;
    }
}
```

```
package examples.nbank;

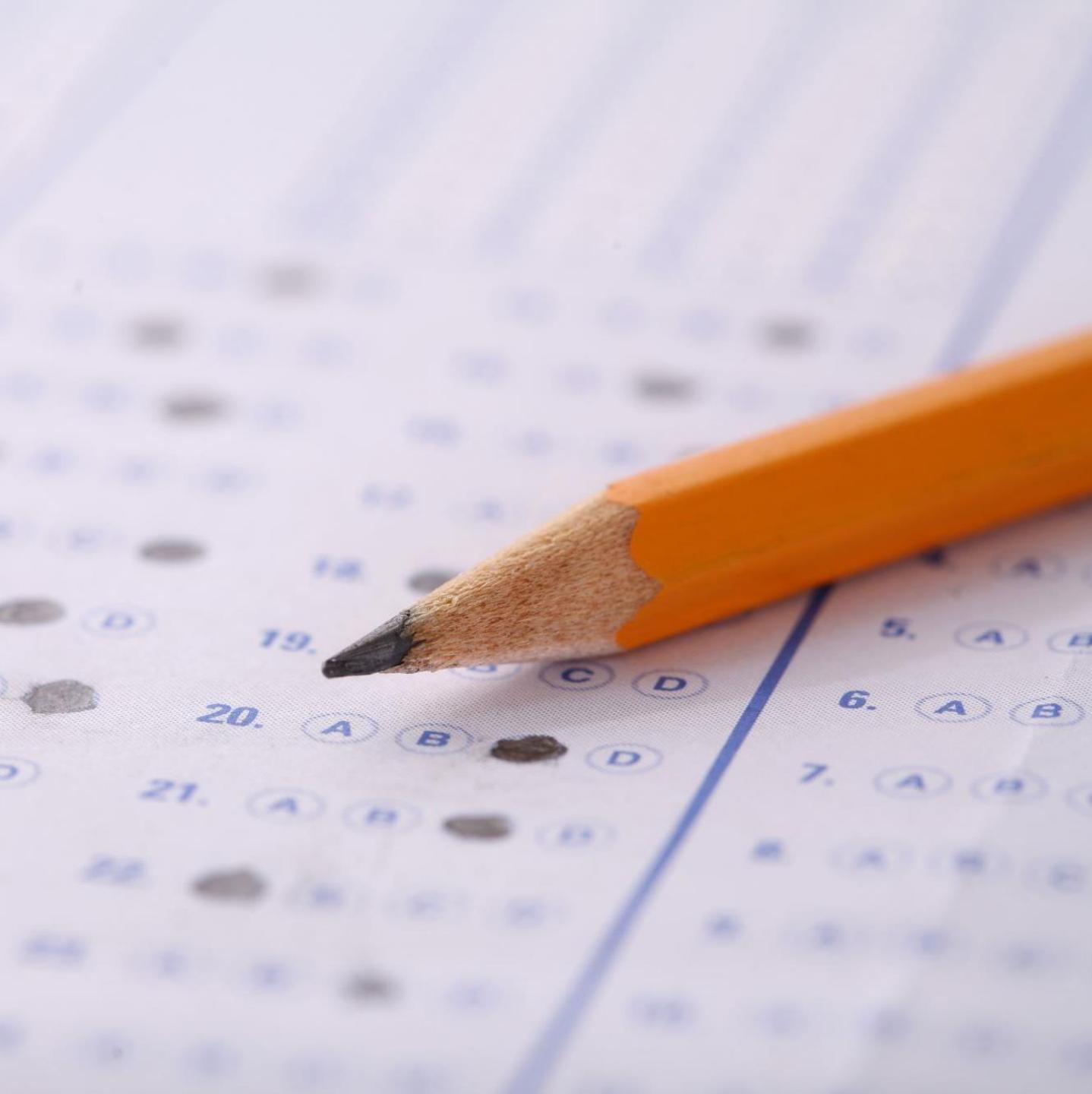
import static org.junit.Assert.assertEquals;
import org.junit.*;

public class ConversionTest {

    @Test
    public void testTempConversion() throws Throwable {
        // Given
        Conversion underTest = new Conversion();

        // When
        double temperature = 80.0d;
        String unit = "";
        double result = underTest.tempConversion(temperature, unit);

        // Then - assertions for result of method tempConversion(double, String)
        assertEquals(176.0d, result, 0.0);
    }
}
```



**TRABAJANDO
CON TEST**



EXCEPCIONES CONTROLADAS

- JUnit nos permite comprobar que un método lanza una excepción controlada.
 - Deben extender de Throwable. Por ejemplo RuntimeException.
 - `assertThrows(Exception.class, () -> {});`
 - `assertThrows(Exception.class, () -> {}, message);`

Assert all

- Difícil seguimiento de asserts cuando hay muchos en un test.
- Si un assert falla, no se ejecutan los siguientes y no sabemos su evaluación.
- `assertAll` ejecuta todos los assert independientemente del posible fallo de uno de ellos.
- Reporta todos los fallos. Dónde se han producido y por qué.



Ejemplo de uso

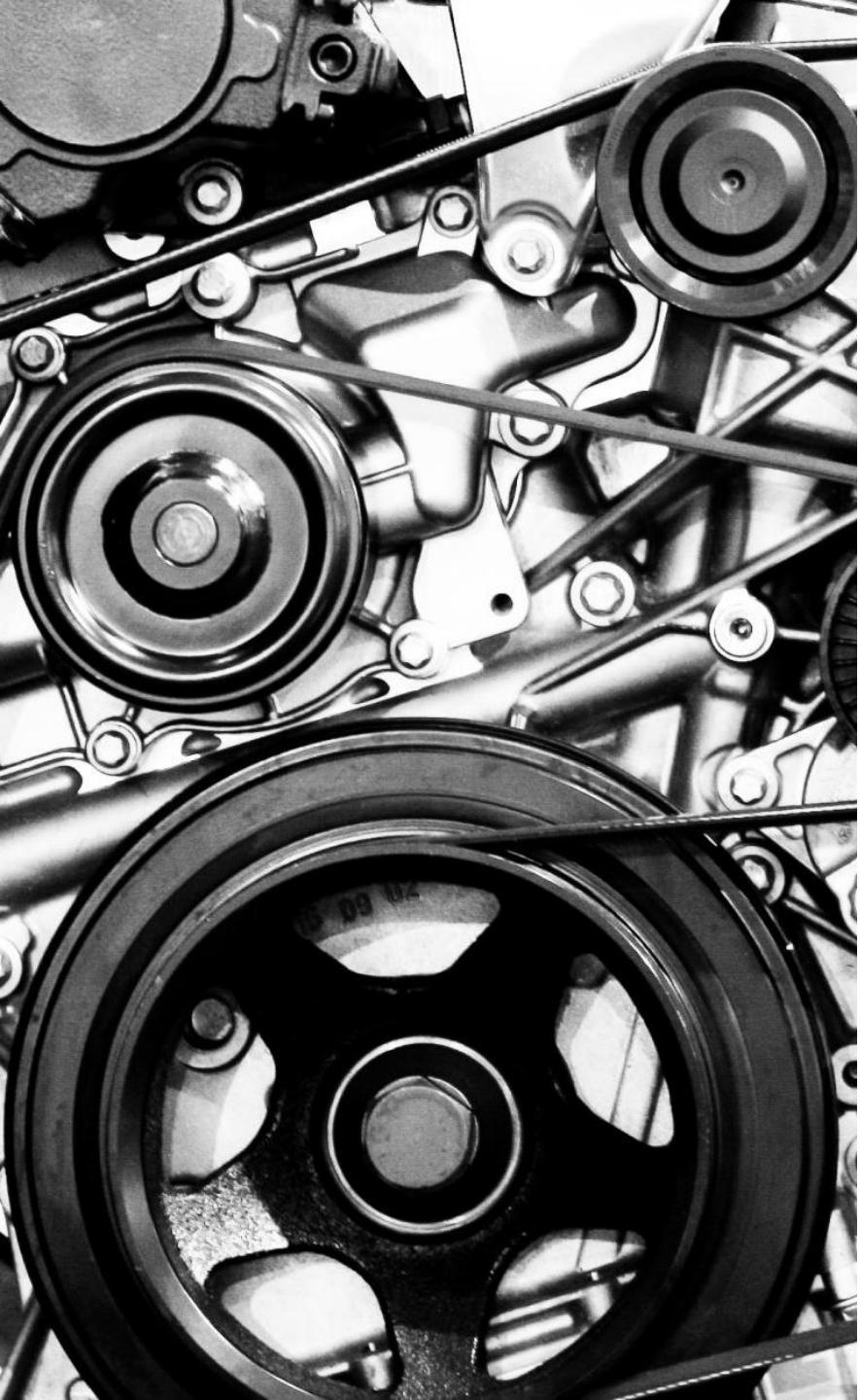
- `assertAll(String message, () -> {});`
- Pueden incluir varias expresiones lambda.
- `assertAll("probando compras", () ->
assertNotNull(store.getProducts()), () ->
assertEquals(2,
store.getProducts().size()), () ->
assertTrue(new
BigDecimal("300.00").compareTo(actual
Amount) == 0));`





Dar nombre a los tests

- JUnit 5 nombra a los tests con el nombre del método.
- Permite especificar un nombre personalizado anotando el método con `@DisplayName("Nombre del test")`;
 - `@Test`
 - `@DisplayName("Comprobación suma calculadora")`
 - `void sumaCalculadora() { ... }`



CICLO DE VIDA

- Proceso por el cual se crea, se ejecuta y se destruye una instancia encargada de la realización de las pruebas
- Se encarga el motor de JUnit 5.
- Se crea una nueva instancia con cada test que se ejecuta.
- JUnit 5 permite ejecutar hooks en diferentes momentos del ciclo de vida.
- Hooks de JUnit 5: @BeforeAll / @AfterAll
@BeforeEach/ @AfterEach

@BeforeEach / @AfterEach

- Se ejecuta una vez que se crea una nueva instancia, es decir, cada vez que se ejecuta un test.
- @BeforeEach - Se ejecuta antes de la ejecución del test.
- @AfterEach - Se ejecuta después de la ejecución del test.

```
public class TiendaTest {  
    private List<String> products;  
@BeforeEach  
    void setup() {  
        products = Arrays.asList("product1",  
"product2");  
    }  
  
@AfterEach  
    void teardown() {  
        products.clear();  
    }  
}
```



@BeforeAll/@AfterAll

- Se ejecuta antes de crear/después de destruir la instancia, por lo que se implementa en un método estático.
- Si se anota en un método no estático, éste fallará, ya que la instancia no existe.
- Se puede forzar que la instancia sólo se cree una vez, aunque es mala práctica, pues compartes el estado de la clase entre tests.
- @Creando una instancia por clase, nos permite quitar el static a @BeforeAll/@AfterAll



DESHABILITANDO TESTS UNITARIOS

- Los tests pueden deshabilitarse para evitar su ejecución.
- Anotamos el test con `@Disabled`
 - JUnit 5 recomienda especificar un motivo:
- `@Disabled("Se deshabilita este test hasta que el bug @B54 se resuelva")`

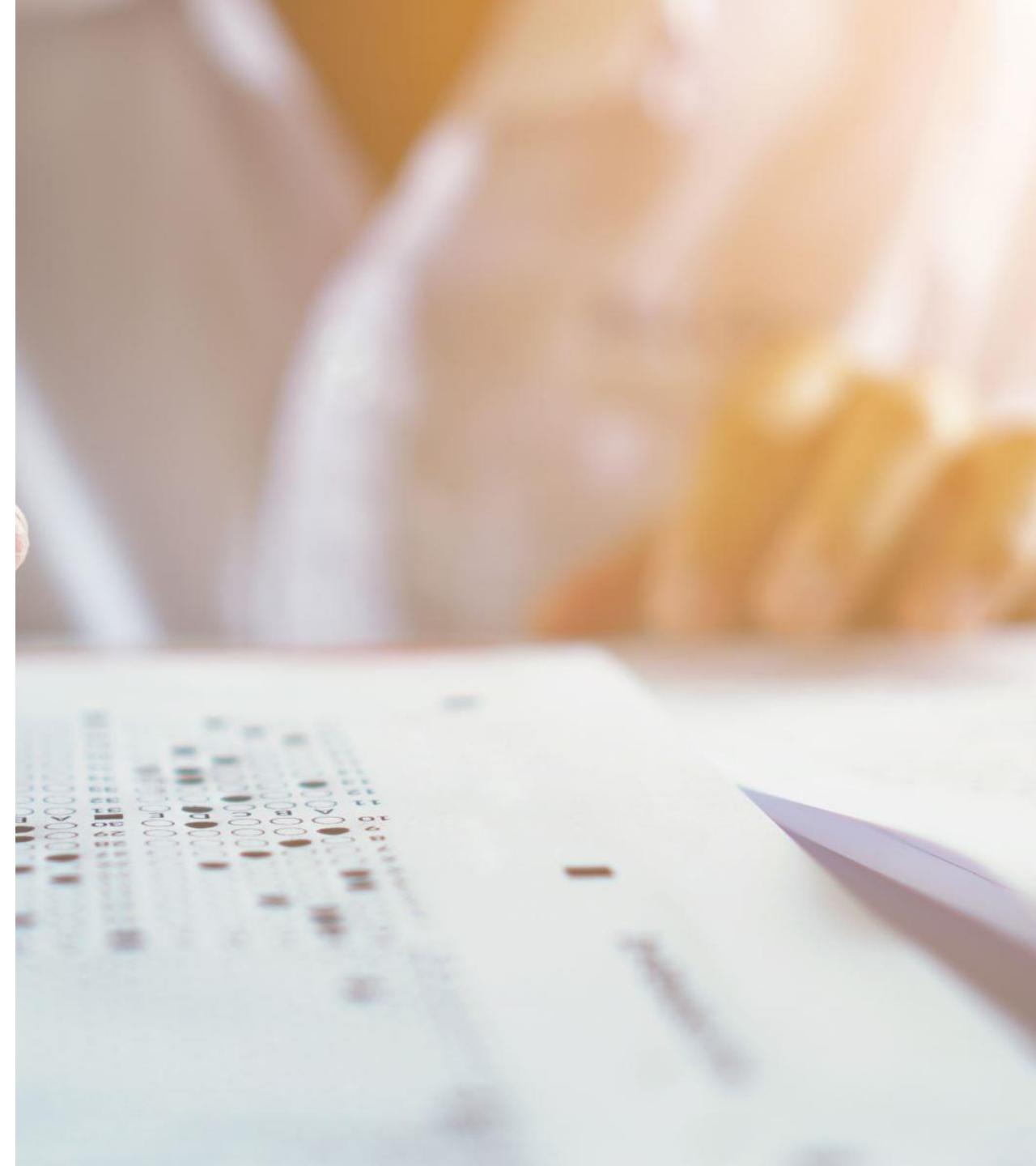
TESTS CONDICIONALES - ANOTACIONES

- Las pruebas unitarias se pueden ejecutar conforme a diferentes condiciones.
- JUnit ofrece anotaciones para habilitar o no dichos tests:
 @EnabledOnOS @EnabledOnJre @EnabledIfSystemProperty
 @EnabledIfEnvironmentVariable

```
@EnabledIf("hasStock")
```

```
@Test
```

```
void it_should_decrease_stock(){  
}  
  
boolean hasStock(){  
}
```



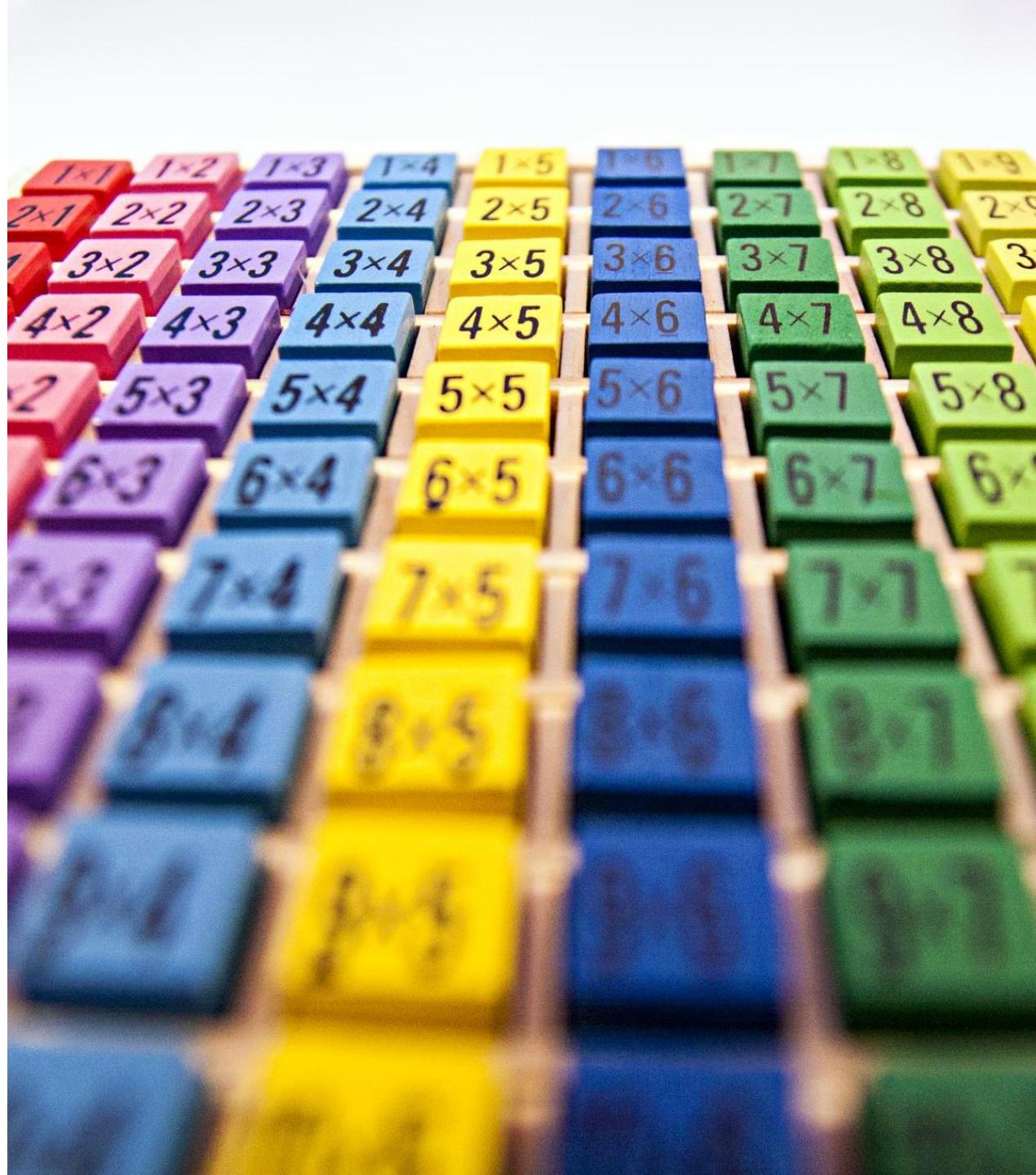
TESTS CONDICIONALES

- Permiten habilitar parte de un test en función de si se cumple una condición. `assumeTrue(boolean condition);`
`assumeFalse(boolean condition);`
- Si la assumption no se cumple, el código a partir de ahí se deshabilita, evitando el fallo
- JUnit 5 permite ejecutar o no parte del método encapsulándolo en una expresión lambda. ○ `assumingThat(boolean, () -> {});`
- La expresión lambda sólo se va a ejecutar si la expresión evalúa a true.
- El código fuera de la expresión lambda, sí se ejecutará.



Clases anidadas

- Las clases anidadas permiten organizar los tests por diferentes criterios: funcionalidad, condicionalidad... Se anotan las clases con @Nested.
- Se puede incluir una descripción en dichas clases y los métodos que contiene con @DisplayName. Los tests aparecerán en el reporting agrupados por clases. Si falla un test de una clase @Nested, aparecerá como fallo el test y las clases contenedoras del mismo (hasta la clase raíz).



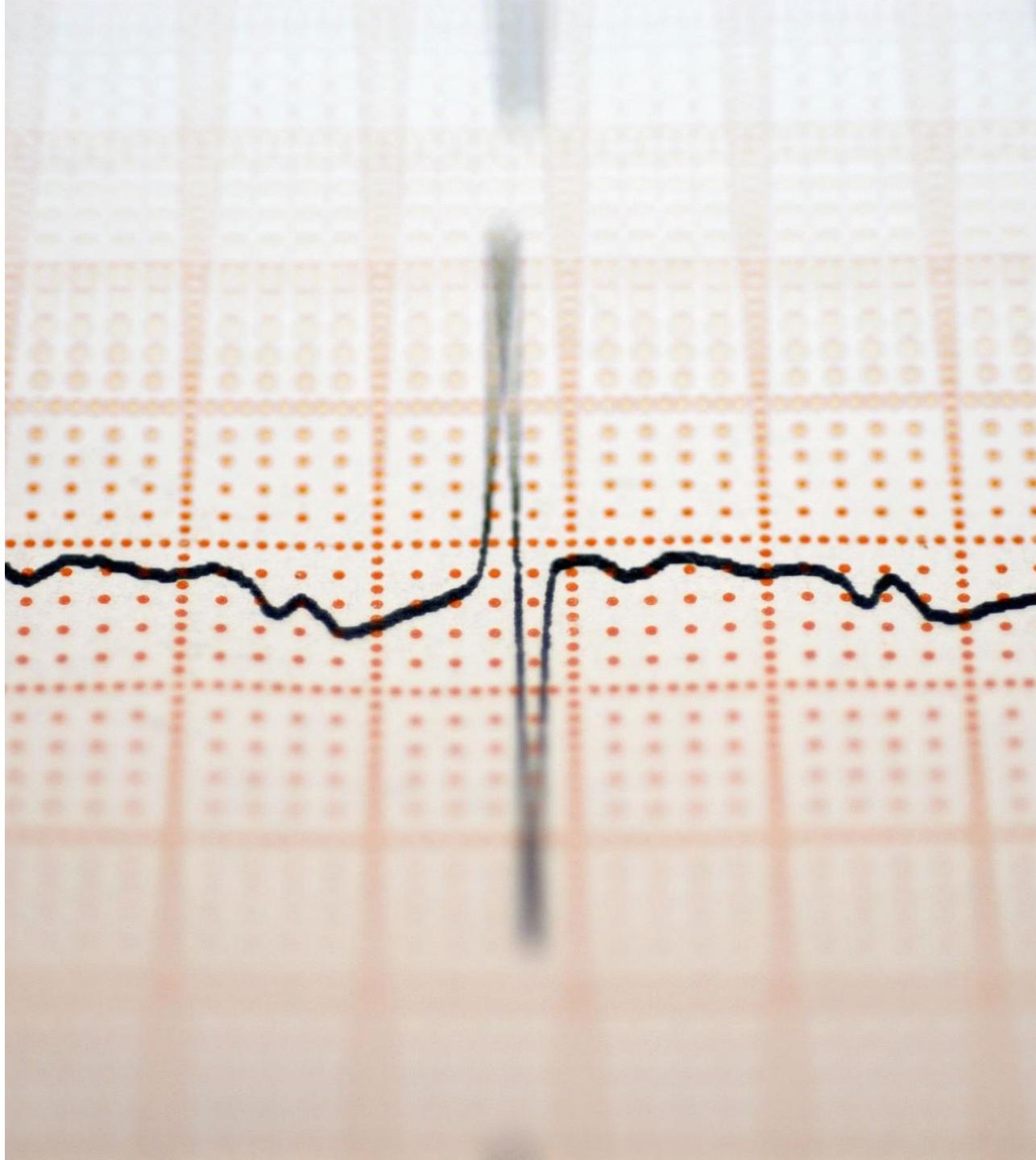
Repetir tests

- JUnit 5 permite ejecutar varias veces el test. Útil en métodos que presentan cierta aleatoriedad.
 - Por ejemplo, crean valores random. Se anotan con @RepeatedTest(int repetitions)
- En el reporting, aparece la ejecución de las repeticiones. Es personalizable el nombre en cada ejecución del test.
@RepeatedTest(int, message)
- Podemos usar variables para la creación de ese mensaje:
{currentRepetition}{totalRepetitions}
- Podemos combinar el nombre con @DisplayName.
@DisplayName: Será el título principal. @RepeatedTest: Nombre en cada repetición. @DisplayName puede ser inyectado en el mensaje de @RepeatedTest:
 - {displayName}



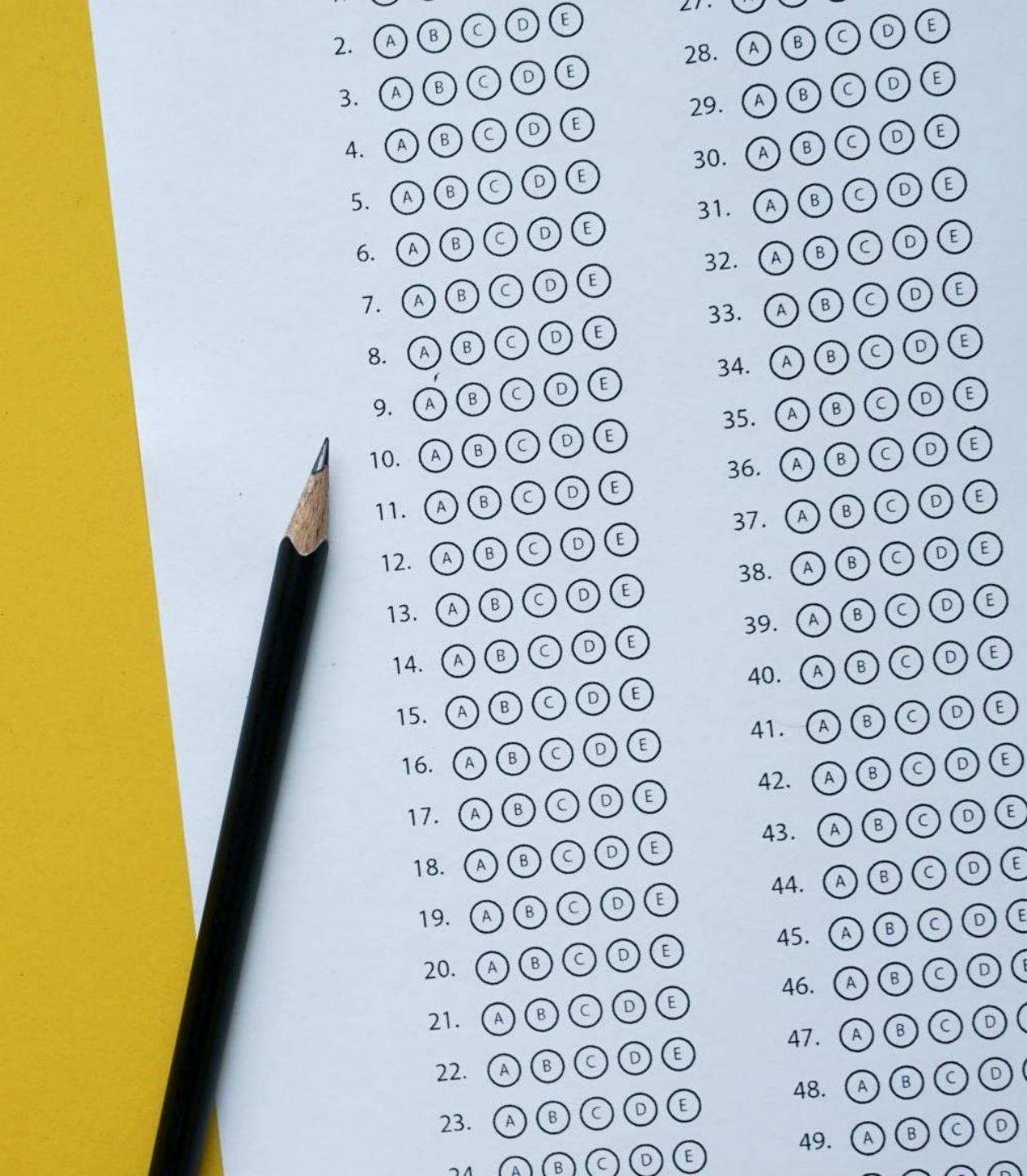
TESTS PARAMETRIZADOS

- Otra forma de repetir tests en JUnit 5.
- Permite que en cada repetición se ejecute con datos diferentes. Se inyectan mediante variables en los métodos.
- JUnit 5 permite proporcionar dichos datos mediante: `@ValueSource(strings={})` Otros tipos: ints, doubles, booleans.
`@CsvSource({índice, valor})`
`@CsvFileSource(resources, delimiter, numLinesToSkip)` `@MethodSource(static methodName)`



Filtrar tests

- Permite ejecutar los tests de forma selectiva.
- Útil para identificar un test con el id de una tarea. Posibilidad de ejecutar grupos de tests.
- Se anota el test o la clase con @Tag. Se puede anotar con varios @Tag.
- Se especifican en el RunConfiguration / Maven.



EJERCICIO 1

```
▶ packages
  ▶ scripts
  ▶ src
  ▶ test
  ▶ types
  .babelrc.js
  .editorconfig
  .eslintignore
  .eslintrc.js
  .flowconfig
  .gitignore
  BACKERS.md
  LICENSE
```

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more attributes (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template compiler

Selenium IDE: Comandos Básicos

- Selenium IDE permite grabar y reproducir pruebas de forma sencilla.
- Es una excelente herramienta para principiantes en automatización.
- Los comandos básicos incluyen abrir, clickear y verificar elementos.





Definir las pruebas de
aceptación



Pruebas de
aceptación



Aplicar las pruebas de
aceptación



Corregir, definir
nuevas historias



Selenium



Selenium IDE



Selenium
WebDriver



Selenium Grid

LABORATORIO 2

```
▶ packages
  ▶ scripts
  ▶ src
  ▶ test
  ▶ types
  .babelrc.js
  .editorconfig
  .eslintignore
  .eslintrc.js
  .flowconfig
  .gitignore
  BACKERS.md
  LICENSE
```

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more attributes (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template compiler



Selenium RC: Importación desde Selenium

- Selenium RC permite la automatización de pruebas en múltiples navegadores.
- Facilita la importación de scripts desde Selenium IDE.
- Es esencial para pruebas más complejas que requieren interacciones personalizadas.



Ejecución de Script en Selenium: Core desde Eclipse

- Eclipse es un entorno de desarrollo integrado para Java.
- Permite ejecutar scripts de prueba de Selenium directamente.
- Facilita la gestión y edición de pruebas automatizadas.

Downloads

Below is where you can find the latest releases of all the Selenium components.
You can also find a list of previous releases, source code, and additional information for Maven users.

Selenium Clients and WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on GitHub.



C#

Stable: [4.13.1 \(September 25, 2023\)](#)

[Changelog](#)

[API Docs](#)



Ruby

Stable: [4.13.1 \(September 25, 2023\)](#)

[Changelog](#)

[API Docs](#)



Java

Stable: [4.13.0 \(September 25, 2023\)](#)

[Changelog](#)

[API Docs](#)



Python



JavaScript

Selenium Test Cases - Demo_Test/Test.java - Eclipse IDE

File Edit Source Refactor Search Project Run Window Help

Quick Access

Package Explorer

Demo_Test

src

(default package)

First.java

JRE System Library [JavaSE-1.8]

Referenced Libraries

client-combined-3.13.0.jar - D:\backups\Selenium

client-combined-3.13.0-sources.jar - D:\backups\Selenium

byte-buddy-1.8.3.jar - D:\backups\Selenium Java

commons-codec-1.14.jar - D:\backups\Selenium

commons-exec-1.3.jar - D:\backups\Selenium Java

commons-logging-1.2.jar - D:\backups\Selenium

guava-23.0-jre.jar - D:\backups\Selenium Java\libs

guava-23.0.jar - D:\backups\Selenium Java\libs

httpclient-4.5.5.jar - D:\backups\Selenium Java\libs

httpcore-4.4.9.jar - D:\backups\Selenium Java\libs

okhttp-3.10.0.jar - D:\backups\Selenium Java\libs

okio-1.14.1.jar - D:\backups\Selenium Java\libs

First.java (1)

```
1 public class First {
2
3     public static void main(String[] args) {
4         // TODO Auto-generated method stub
5     }
6
7 }
8
9
10 }
```

Task List

Find All Activate

Outline

First

mainString(): void

Problems

JavaDoc Declaration

0 items

Description	Resource	Path	Location	Type

A screenshot of a web browser window displaying the Spanish Wikipedia homepage (es.wikipedia.org/wiki/Wikipedia:Portada). The URL bar at the top is highlighted with a red arrow. The page content includes the Wikipedia logo, a welcome message in Spanish, and a sidebar with various links. A developer tools panel is open over the page, specifically the 'Elements' tab, which shows the HTML structure of the page's header, including the title and meta charset information.

```
<!DOCTYPE html>
<html class="client-js ve-not-available" lang="es" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <title>Wikipedia, la enciclopedia libre</title>
```

Variación	Descripción	Muestra
By.className	Encuentra elementos basados en el valor del atributo «class»	findElement(By.className(«algúnnombrecrase»))
By.cssSelector	Encuentra elementos basados en el motor selector CSS subyacente del controlador	findElement(By.cssSelector(«input#email»))
By.id	localiza los elementos por el valor de su atributo «ID»	findElement(By.id(«algunID»))
By.linkText	Encuentra un elemento link con el texto exacto que muestra	findElement(By.linkText(«AlgunTexto»))
By.name	Localiza los elementos por el valor del atributo «Name»	findElement(By.name(«AlgunName»))
By.partialLinkText	Localiza elementos que contienen el texto de enlace dado	findElement(By.partialLinkText(«REG»))
By.tagName	Localiza elementos por su nombre de etiqueta	findElement(By.tagName(«etiqueta»))
By.xpath	Localiza elementos mediante XPath	findElement(By.xpath(«rutacpath»))

The screenshot shows a web browser window with the Wikipedia homepage loaded. The left sidebar contains links like 'Portada', 'Portal de la comunidad', 'Actualidad', 'Página aleatoria' (highlighted with a red arrow), 'Ayuda', 'Donaciones', 'En otros proyectos', 'Wikimedia Commons', 'MediaWiki', and 'Meta-Wiki'. The main content area has a heading 'Bienvenido a Wikipedia' and a list of navigation links. The developer tools' Elements tab is open, displaying the HTML structure of the page. A red arrow points from the 'Página aleatoria' link in the sidebar to its corresponding element in the DOM tree. Another red arrow points from the 'n-randompage' ID in the DOM tree to the line of code in the bottom right corner.

```
driver.findElement(By.id("n-randompage")).click();
```

```
* <div class="portal" role="navigation" id="p-navigation" aria-labelledby="p-navigation-label">
  <h3 id="p-navigation-label">Navegación</h3>
  <div class="body">
    <ul>
      <li id="n-mainpage-description"></li>
      <li id="n-portal"></li>
      <li id="n-currentevents"></li>
      <li id="n-recentchanges"></li>
      <li id="n-newpages"></li>
      <li id="n-randompage" style="background-color: #e0f2e0; border: 1px solid #80c0ff; border-radius: 5px; padding: 2px 5px; margin-right: 10px;"><a href="/wiki/Especial:Aleatoria" title="Cargar una página al azar [alt-shift-x]" accesskey="X">Página aleatoria</a></li>
      <li id="n-help"></li>
      <li id="n-sitesupport"></li>
      <li id="n-log_in_article" style="display: none;"></li>
    </ul>
  </div>
</div>
<div class="portal" role="navigation" id="p-wikibase-otherprojects" aria-labelledby="p-wikibase-otherprojects-label"></div>
```



WIKIPEDIA
La enciclopedia libre

Portada
Portal de la comunidad
Actualidad

86.73 x 14

Página aleatoria

Ayuda

Donaciones

En otros proyectos

Wikimedia Commons

MediaWiki

Meta-Wiki

Wikispecies

Bien

la enciclo
que todo

Artículo de

Roman

Romance

estudio de

Forma pa

y 1978. S

lta~~~~~

```
> <div id="mw-head">.</div>
*<div id="mw-panel">
  <div id="p-logo" role="banner">.</div>
  <div class="portal" role="navigation">
    <h3 id="p-navigation-label">Navegación</h3>
    <div class="body">
      <ul>
        <li id="n-mainpage-description">.</li>
        <li id="n-portal">.</li>
        <li id="n-currentevents">.</li>
        <li id="n-recentchanges">.</li>
        <li id="n-newpages">.</li>
        <li id="n-randompage">
          <a href="/wiki/Especial:Aleatoria" title="Cargar una página al azar [alt-shift-x]" accesskey="x">Página aleatoria</a> -->
        </li>
        <li id="n-help">.</li>
        <li id="n-sitesupport">.</li>
        <li id="n-bug_in_article" style="display: none;">.</li>
      </ul>
    </div>
  </div>
</div>
```

By arg0

driver.findElement(By.linkText("Página aleatoria"))

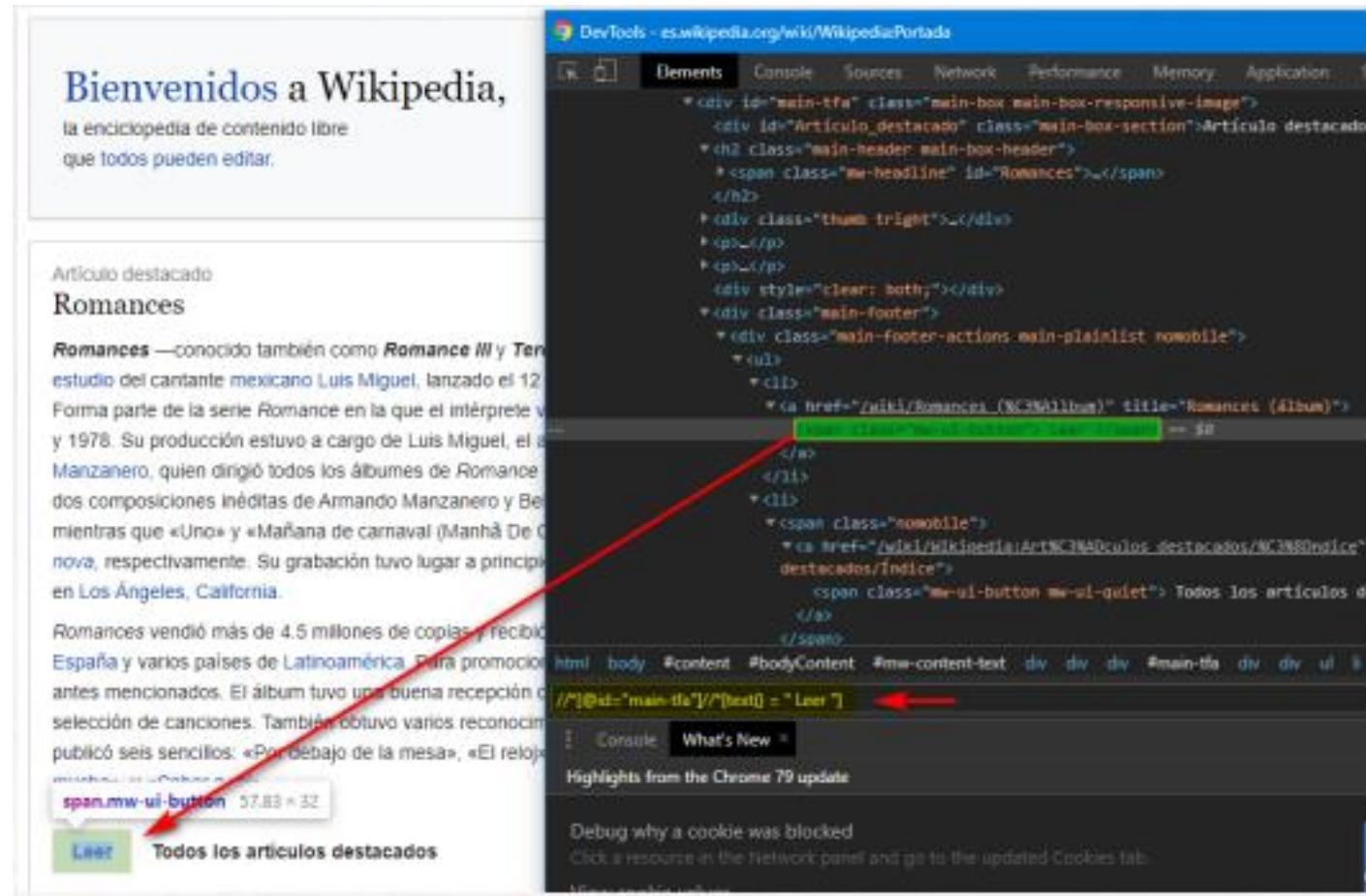
//EXTRACCIÓN DEL TÍTULO

linkText(String linkText) : By - By

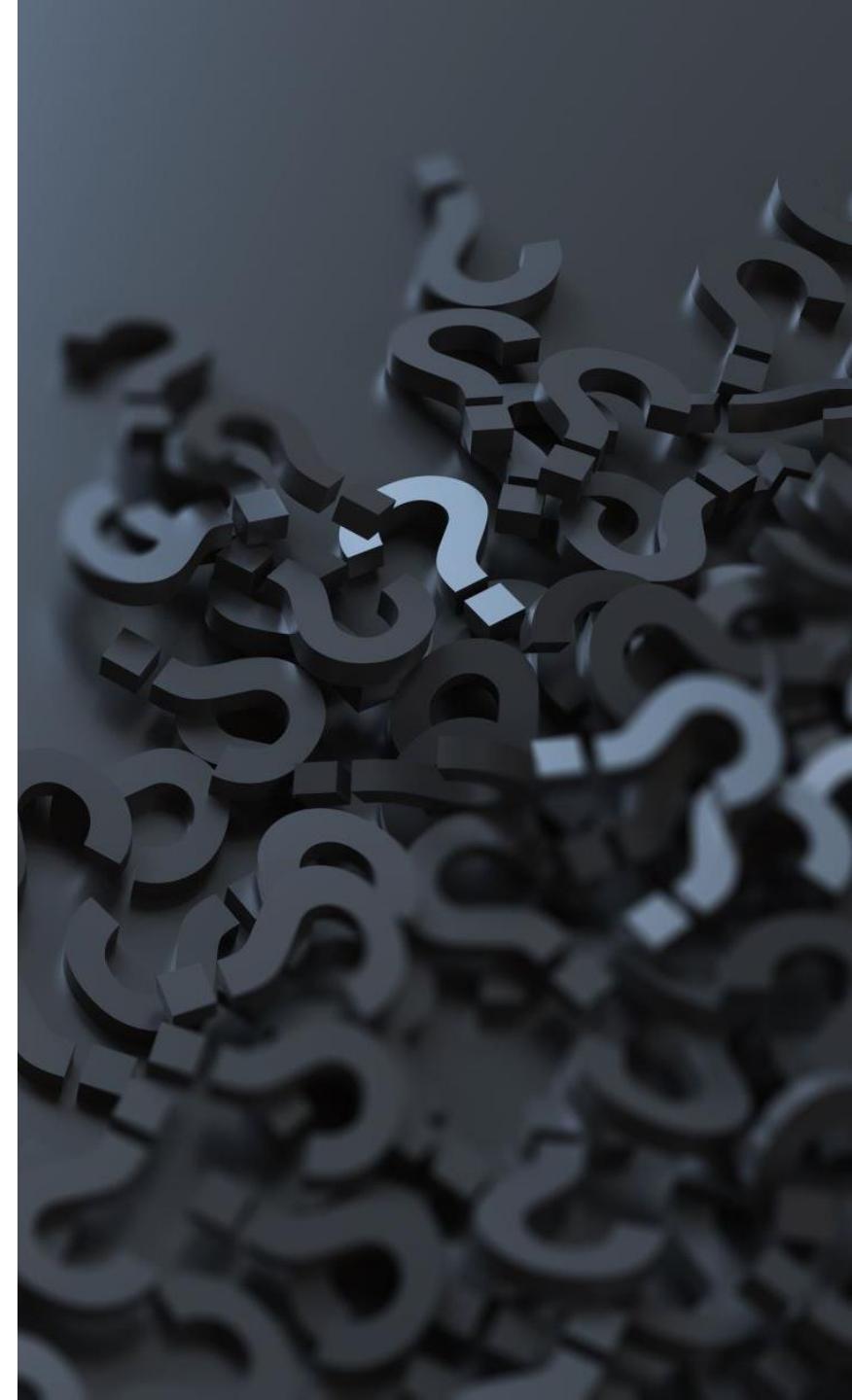


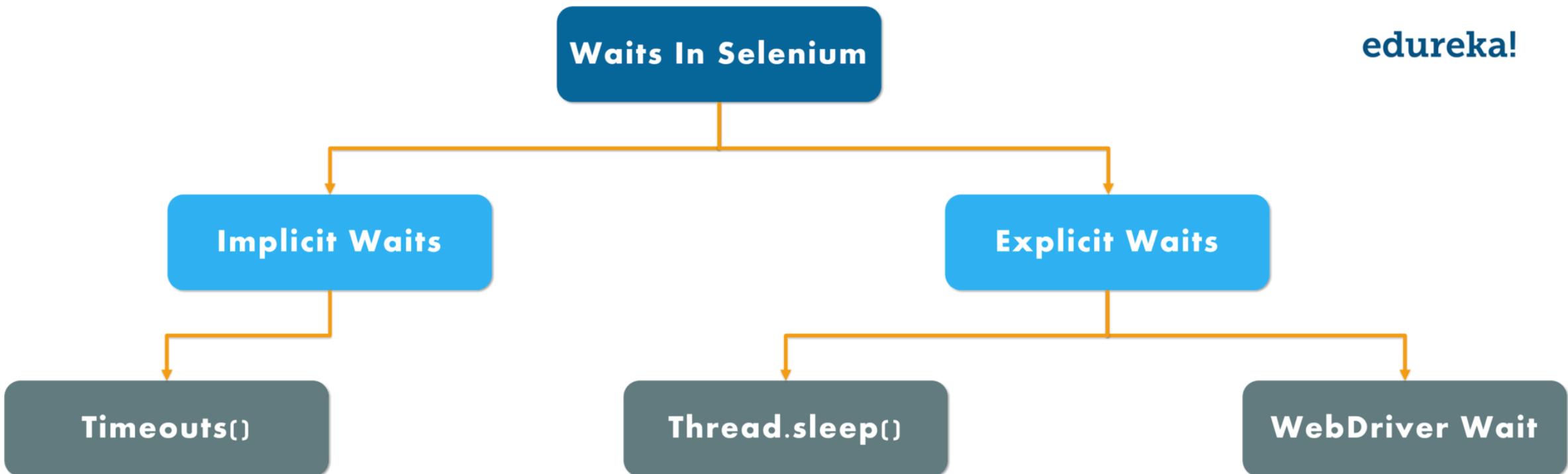
The screenshot shows a browser window with the developer tools open, specifically the Elements tab. On the left, there is a preview of a form with three radio buttons labeled "Carne", "Pescado", and "Verdura". A red arrow points from the "Carne" label to the corresponding radio button in the DOM tree on the right. The DOM tree is as follows:

```
<!DOCTYPE html>
<html>
  <head></head>
  ...<br>
  ...<br>
  ...>
  <input type="radio" name="bebida" value="refresco"> Refresco<br>
  <input type="radio" name="bebida" value="agua"> Aguas<br>
  <input type="radio" name="bebida" value="cafe"> Cafe<br>
  <br>
  <br>
  <br>
  <input type="radio" name="comida1" value="carne">
    " Carne"
    <br>
  <input type="radio" name="comida2" value="pescado">
    " Pescado"
    <br>
  <input type="radio" name="comida3" value="verdura">
    " Verdura"
    <br>
    <br>
  <!--
    <input type="checkbox" name="listaCheckbox" id="checkbox1">Primer valor<br>
    <input type="checkbox" name="listaCheckbox" id="checkbox2">Segundo valor<br>
    <input type="checkbox" name="listaCheckbox" id="checkbox3">Tercer valor<br>
  -->
  </body>
</html>
```



- Las páginas webs cargan sus recursos de manera asíncrona.
- Esto implica que los elementos no cargan en el mismo orden ni al mismo tiempo siempre.
- Debido a ello, nosotros necesitamos implementar mecanismos de espera que aguarden a que el elemento que busquemos termine de cargar correctamente.
- En caso de no implementar dichos mecanismos, si Selenium no encuentra el elemento, devolverá un error “NoSuchElementException”.
- Aunque el primer impulso sea detener la ejecución del hilo durante un tiempo determinado “Thread.sleep(1000)”, esta manera de hacerlo es incorrecta.







JUnit

LABORATORIO 3

```
▶ packages
  ▶ scripts
  ▶ src
  ▶ test
  ▶ types
  .babelrc.js
  .editorconfig
  .eslintignore
  .eslintrc.js
  .flowconfig
  .gitignore
  BACKERS.md
  LICENSE
```

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

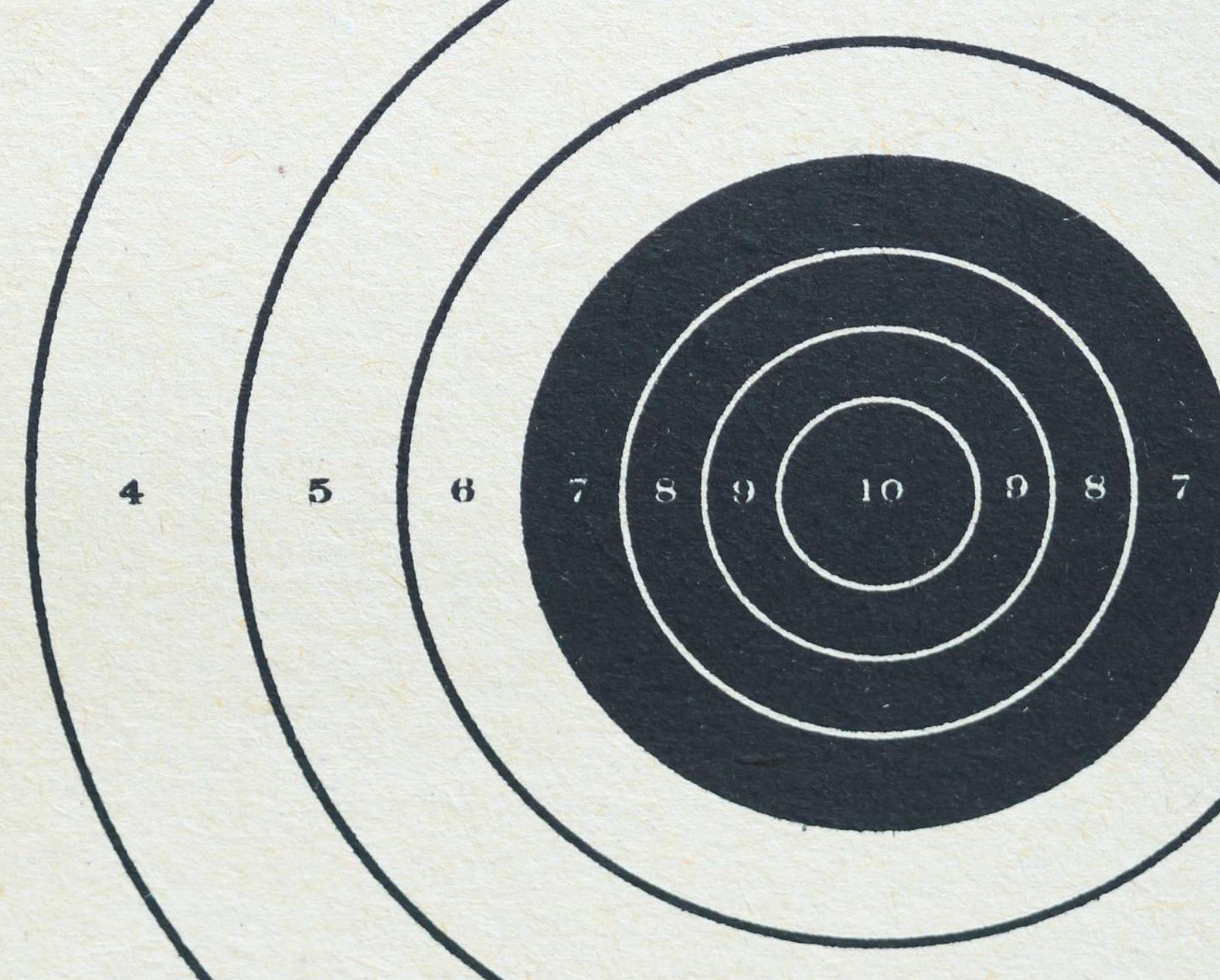
refactor: extend dom-props update skip to more attributes (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template compiler

DEMO



EJERCICIO

2

node_modules
bower_components
flowconfig
packages
scripts
src
test
types
.babelrc.js
.editorconfig
.eslintignore
.eslintrc.js
.flowconfig
.gitignore
BACKERS.md
LICENSE

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more attributes (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify timestamp check



Mockito: Conceptos Básicos

- Mockito es un marco de simulación para pruebas unitarias en Java.
- Permite crear objetos simulados para pruebas más efectivas.
- Facilita la validación del comportamiento de las interacciones.

\$\$\$

Tiempo de desarrollo

₡

E2E (5%)

Integración (APIs
o *Backend*) (15%)

Pruebas unitarias (80%)



Tiempo de ejecución





Unit testing



Integration testing



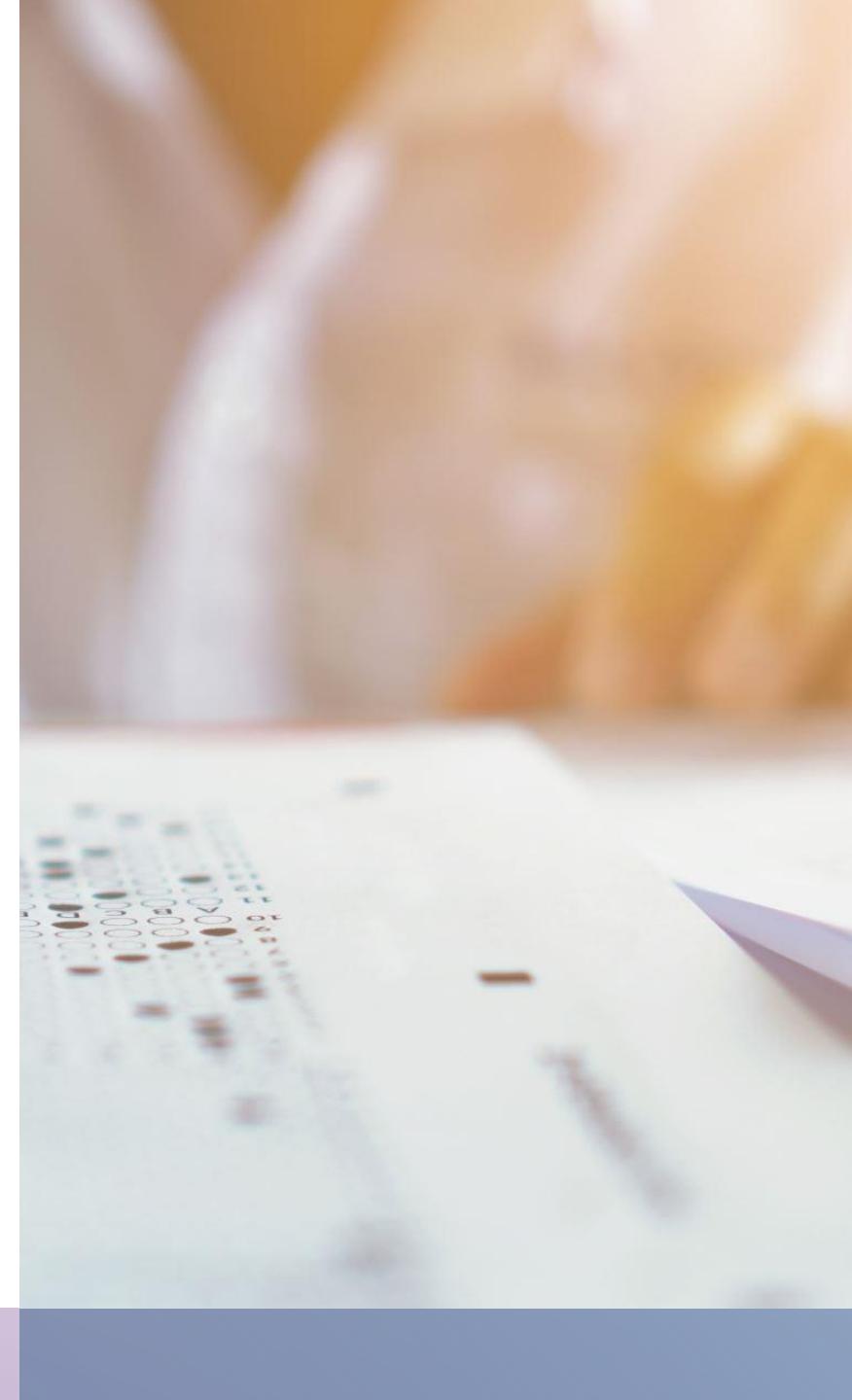
Functionality testing

mockito



¿POR QUÉ UTILIZARLO?

- Permite escribir y ejecutar tests unitarios de código integrado por varios componentes.
- Simula el comportamiento de componentes (mock, spy).
- Proporciona rapidez en las pruebas.
- Útil para seguir el paradigma TDD/BDD.
- Ejemplos de uso: Conexiones con BD. Servicios web. Clases de lenta ejecución. Clases con side-effects. Clases con un comportamiento indefinido.



MOCKITO BDD



Mockito permite basarnos en el comportamiento de otros componentes:



Dado que (Given): Preparación estado inicial.



Cuando (When): Se invoca al método.



Entonces (Then): Validamos el comportamiento esperado

PRIMEROS TESTS

```
@Test  
Run Test | Debug Test | ✓  
void our_first_test_with_mockito() {  
    // Creamos el objeto ficticio.  
    List<String> first_mock = mock(ArrayList.class);  
  
    // Simulamos el comportamiento  
    when(first_mock.get(0)).thenReturn("first_element");  
  
    String first_element = first_mock.get(0);  
  
    // Verificamos  
    assertEquals("first_element", first_element);  
    verify(first_mock).get(0);  
}
```

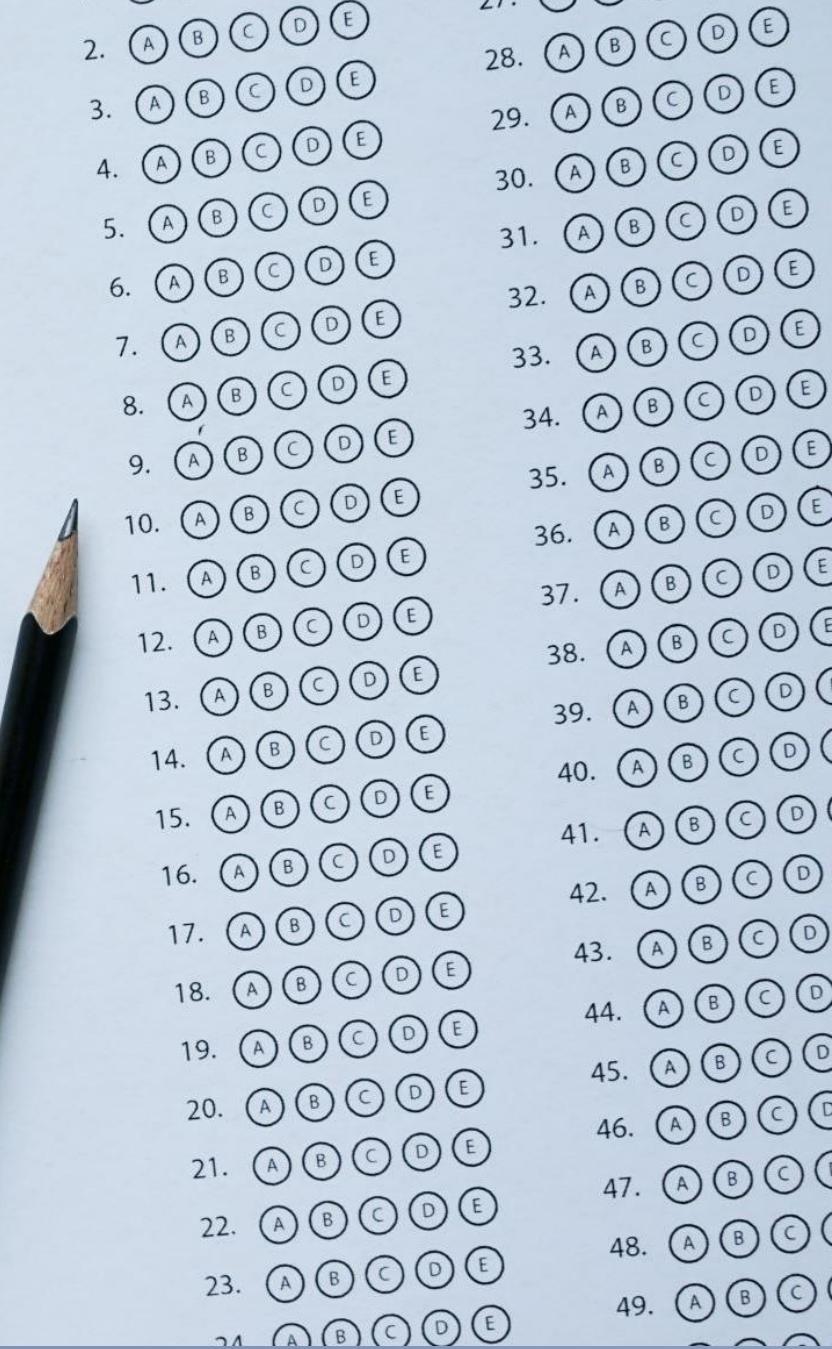
ARGUMENT MATCHERS

- Los ArgumentMatchers de Mockito permiten:
- Verificar argumentos con los que son llamados métodos de los mocks.
- Simular el comportamiento de un componente.
- eq() any(), anyInt(), anyString()... isNull(), isNotNull(), isA(Class c)

A blackboard filled with mathematical calculations and formulas. At the top, there are several equations involving summation and square roots. In the center, there is a diagram of a circle with radius 'c'. Below it, there is a right triangle with legs labeled 'x' and 'y'. To the right of the triangle, there is a system of equations: $xy = c$, $cx - cy = 0$, and $2\pi = c$. Further down, there is a complex fraction: $\frac{4x}{y} + \frac{x^2 + y^2}{c}$, which is simplified to $= 584. + n^{av}$. To the right of this, there is a formula involving a sum: $\sum N^{30} \cdot x - \frac{1}{2} [g_{64} + x]$. At the bottom left, there is a diagram of a circle with radius 'r=4'. To the right of the circle, there is a formula: $\beta = 9 + x^2$.

VERIFY

- Método estático.
- Permite testear la ejecución de un mock.
- `verify(mock, [times(int number)]).método()`
- `verify(mock, never()).método()`
- `verifyNoInteractions()`
- `verify(mock, {atLeast(int), atMost(int), atLeastOnce(), atMostOnce()})`



MOCK VS SPY



LABORATORIO 4

v2.6.0-beta.2 · build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

origin/dynamic-directive-arguments

feat: dynamic args for customizations

perf: improve scoped slots change detection accuracy (#9371)

RIO 4 for v-only-bind dynamic arguments

refactor: y-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots changes

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more sinks

fix: fix checkbox event edge case in Firefox

test· fix tests in IE/Edge

refactor: simplify threads and code

```
▶ src
  ▶ package
  ▶ scripts
  ▶ src
  ▶ test
  ▶ types
```

.babelrc.js

editorconfig

`eslintignore`

`.eslintrc.js`

flowconfig

PACKERS md

LICENSE

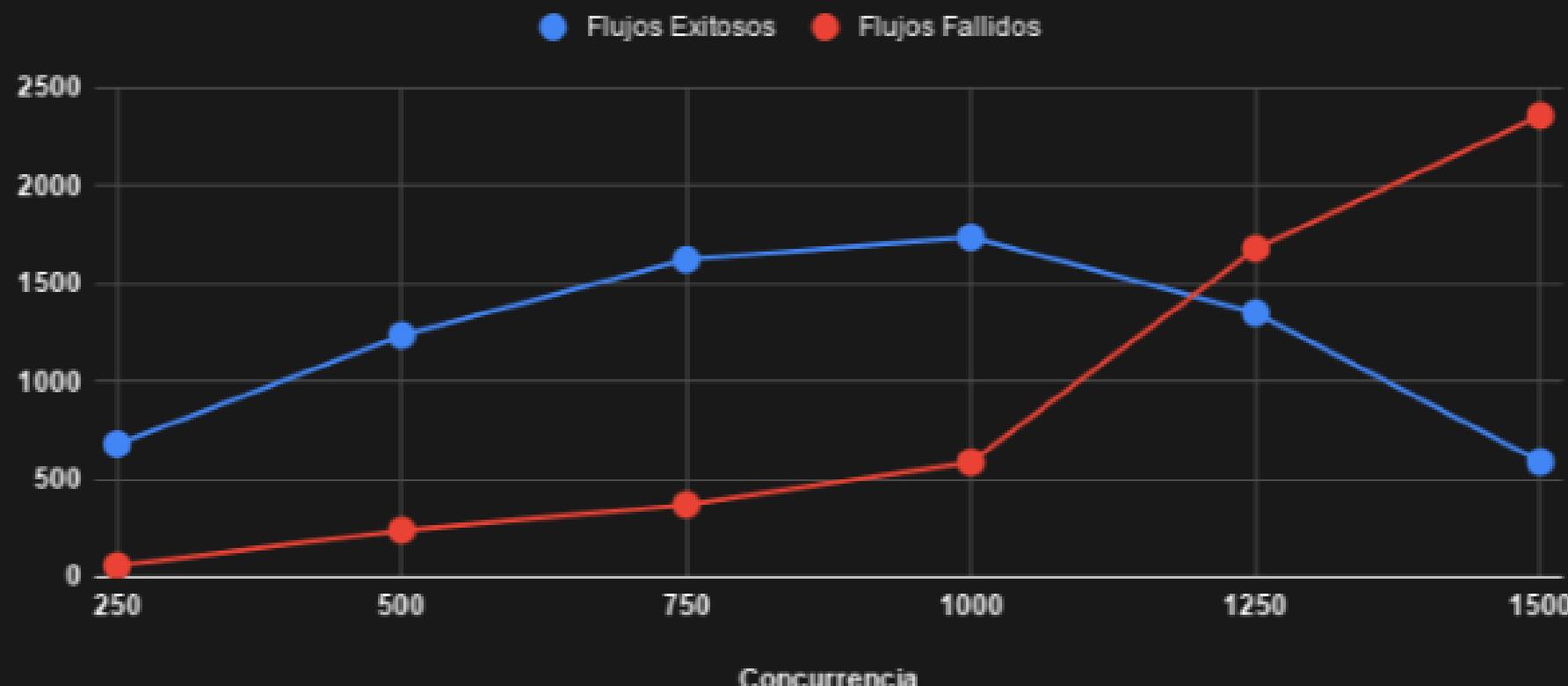
JMeter (Básico): Elementos Básicos

- JMeter es una herramienta de código abierto para pruebas de rendimiento.
- Permite grabar y ejecutar scripts de pruebas de carga.
- Utiliza diferentes elementos como hilos, grupos y controladores.

Punto de Inflección

De acuerdo a los resultados de las diferentes cargas realizadas, se determina el punto de quiebre de la plataforma, analizando los flujos exitosos vs los fallidos.

Punto de Inflección



TIPOS

- **Pruebas de carga:** Evalúan cómo responde el sistema bajo una cantidad específica de solicitudes.
- **Pruebas de estrés:** Prueban el sistema con una carga mayor a la esperada para encontrar el punto de fallo.
- **Pruebas de escalabilidad:** Analizan cómo se comporta el sistema al aumentar gradualmente la carga.
- **Pruebas de latencia:** Miden el tiempo de respuesta de operaciones específicas.
- **Pruebas de memoria:** Detectan fugas de memoria y optimizan el uso de heap



HERRAMIENTAS

- **JMH (Java Microbenchmark Harness):** Framework para microbenchmarks en Java.
- **JProfiler:** Herramienta de análisis de rendimiento y detección de cuellos de botella.
- **VisualVM:** Monitorea la memoria, CPU y ejecución de hilos en tiempo real.
- **YourKit Java Profiler:** Perfilá aplicaciones Java para analizar memoria y rendimiento.
- **Apache JMeter:** Realiza pruebas de carga y estrés en aplicaciones web.
- **Gatling:** Herramienta para pruebas de rendimiento en sistemas web con gran concurrencia.





BlazeMeter Functional Performance Mock Services API Monitoring

Search

Home Demos Projects Tests Reports 0 Create Test SE Demo ⚙

Default project / ATM Search

ATM Search Set Report as a Baseline

Scenario: DBank ATM Search Test-Http Location: US East (Virginia, Google)

Summary Timeline Report End User Experience Monitoring Request Stats Engine Health Failure Criteria Errors Logs Original Test Configuration

21 VU Max Users	18.13 Hits/s Avg. Throughput -27.5% from baseline	69.17 % Errors +64.9% from baseline	1.82 s Avg. Response Time +48.8% from baseline	4.44 s 90% Response Time +50.7% from baseline	26.69 MiB/s Avg. Bandwidth -5.4% from baseline
--------------------	---	---	--	---	--

Duration: 21 minutes Test Type: Selenium, JMeter
Started: Jun 22, 2022, 12:03:19 AM Response Codes: 2xx 4xx
Ended: Jun 22, 2022, 12:29:27 AM Locations: US East (Virginia, Google)

Tags: ATM DBank Report Notes: Enter report notes... Save Note

Load

25 vu 25 vu 2500 ms

22/06/2022 - 05:20:00
• Users: 21
• Hits/s: 19.75
• Error(s): 13.57

Response Time

This screenshot shows the BlazeMeter Performance Test Summary Report for a project named 'ATM Search'. The top navigation bar includes links for Functional, Performance, Mock Services, and API Monitoring, along with a search bar and user profile. The main content area displays key performance metrics: 21 VU (Max Users), 18.13 Hits/s (Avg. Throughput, -27.5% from baseline), 69.17 % Errors (+64.9% from baseline), 1.82 s Avg. Response Time (+48.8% from baseline), 4.44 s 90% Response Time (+50.7% from baseline), and 26.69 MiB/s Avg. Bandwidth (-5.4% from baseline). Below these metrics are detailed logs for duration, start/end times, test type, response codes, locations, and tags. Two line charts at the bottom show 'Load' (Users vs. Time) and 'Response Time' (ms vs. Time), with specific data points highlighted for the end of the test period.

LABORATORIO 5

LICENSE

- ▶ examples
- ▶ flow
- ▶ packages
- ▶ scripts
- ▶ src
- ▶ test
- ▶ types

.babelrc.js

.editorconfig

.eslintignore

.eslintrc.js

.flowconfig

.gitignore

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more situations (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template check

JMeter (Intermedio): Elementos Avanzados



- JMeter permite la creación de pruebas complejas de rendimiento.
- Se pueden generar reportes detallados a partir de las pruebas.
- La ejecución de scripts se puede hacer desde la línea de comandos.

Test Plan

- User Defined Variables
- HTTP Request Defaults
- HTTP Cookie Manager

Thread Group

- View Results Tree

HTTP(S) Test Script Recorder

Name: HTTP(S) Test Script Recorder

Comments:

State

Start Stop Restart

Global Settings

Port: 8888 HTTPS Domains:

Test Plan Creation Requests Filtering

Test plan content

Target Controller: Use Recording Controller

Grouping: Put each group in a new transaction controller Capture HTTP Headers Add Assertions Regex matching

HTTP Sampler settings

Transaction name

Create new transaction after request (ms):

Retrieve All Embedded Resources Follow Redirects

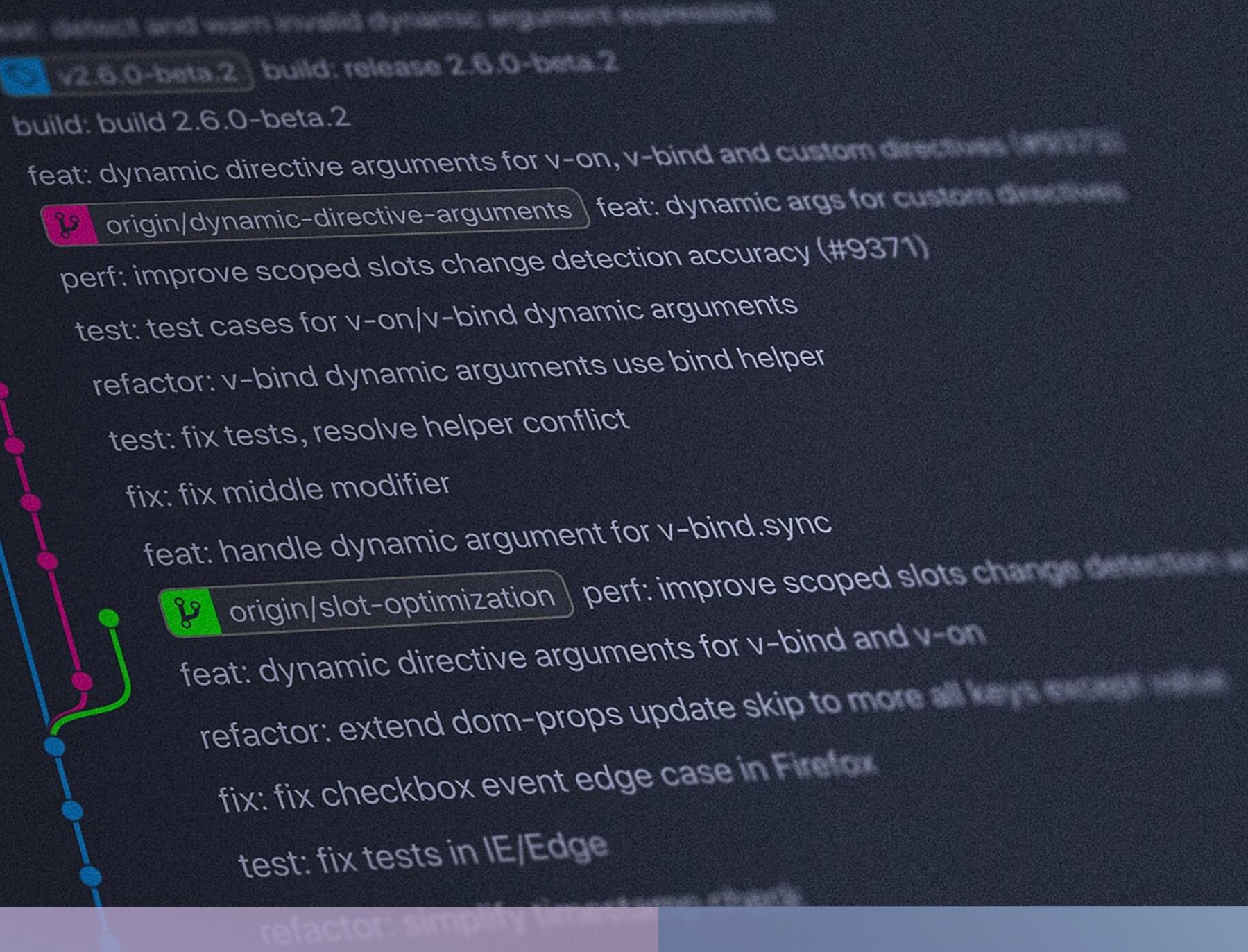
Redirect Automatically

Use KeepAlive

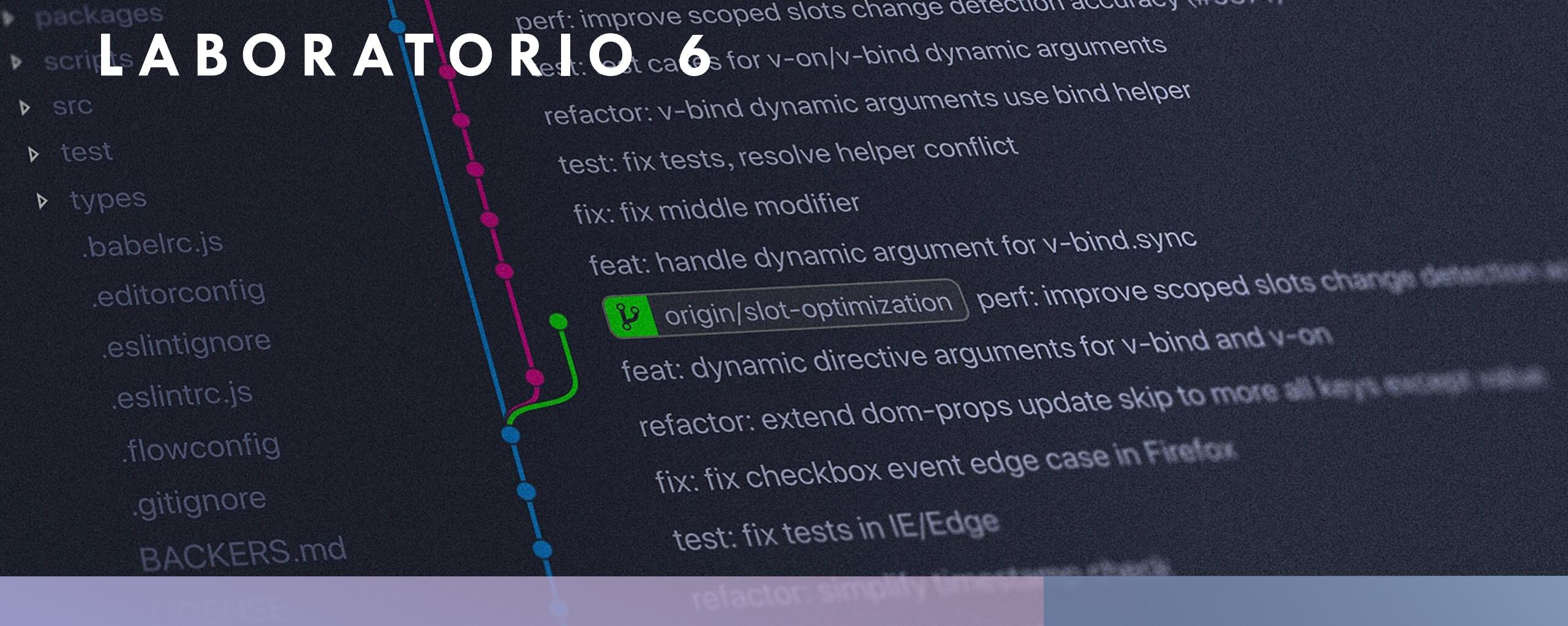
Type:

DEMO

```
▶ packages
  ▶ src
  ▶ test
  ▶ types
  .babelrc.js
  .editorconfig
  .eslintignore
  .eslintrc.js
  .flowconfig
  .gitignore
  BACKERS.md
```



LABORATORIO 6



[Tests & Reports](#)[+ Add Test](#)[? Help](#)

These reports are a result of running a test named: <http://demo.blazemeter.com/drupal6/> on Mon, 06/17/2013 - 13:39.

[Print](#)[Tweet](#)[LOAD RESULTS](#)[ERRORS \(1139\)](#)[MONITORING](#)[REAL BROWSER](#)[PAGE SPEED](#)[LOGS](#)

This report is a result of spawning numerous servers in the cloud, executing a **JMeter** test-script using distributed **JMeter** architecture. All graphs represent KPIs (Key Performance Indicators) measured during the load.

[ALL](#)[1360236271_88e](#)[_user_login](#)[1360236270_88e](#)[1360236745_title](#)[Active Users](#)[Max Users](#)[ALL_Errors](#)[ALL_Latency](#)[Max Users](#)

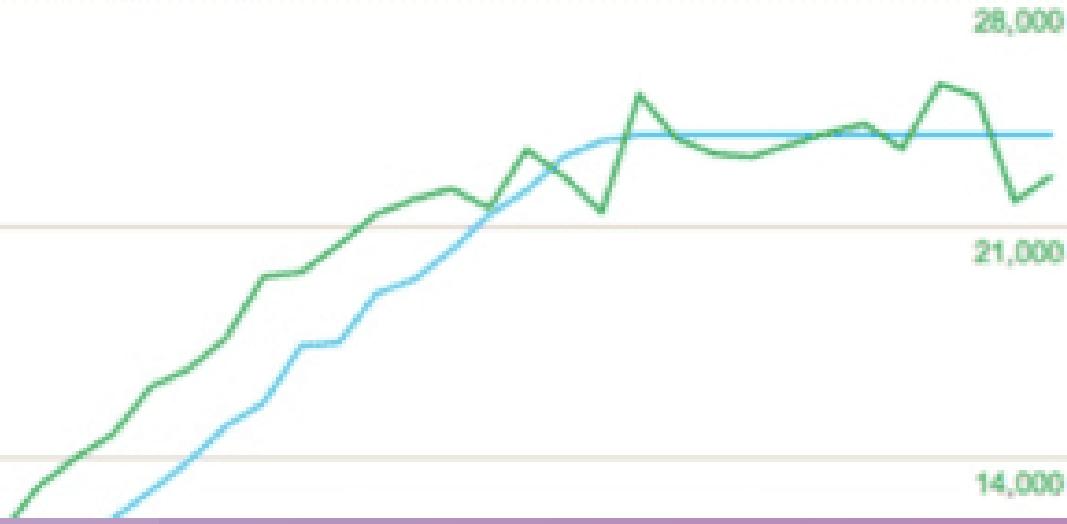
600

28,000

[Response Time](#)[Latency](#)[Hits/s](#)[KB/s](#)[users / errors](#)

450

21,000

[Latency](#)[Hits](#)[KB/s](#)[Support](#)

LABORATORIO 7

```
▶ README.md  
▶ .eslintrc.js  
▶ .editorconfig  
▶ .gitignore  
▶ .babelrc.js  
▶ .eslintignore  
▶ .flowconfig  
▶ .eslintrc.js  
▶ BACKERS.md  
▶ LICENSE  
▶ packages  
▶ scripts  
▶ src  
▶ test  
▶ types
```

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more attributes (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template compiler

Watir: Ejecución de Comandos Básicos

- Watir es una herramienta para la automatización de pruebas web.
- Permite interactuar con navegadores de manera sencilla.
- Los comandos básicos incluyen abrir, clic y llenado de formularios.



CARACTERÍSTICAS

- **Ubicación de elementos web** : existen diferentes formas de ubicar elementos web representados dentro del navegador. Las más utilizadas son id, clase, nombre de etiqueta, atributos personalizados, etiqueta, etc.
- **Capturas de pantalla** : Watir le permite tomar capturas de pantalla de las pruebas realizadas cuando sea necesario. Esto ayuda a realizar un seguimiento de las pruebas intermedias.
- **Rendimiento de la página** : puedes medir fácilmente el rendimiento de la página mediante el objeto de rendimiento, que tiene propiedades como `performance.navigation`, `performance.timing`, `performance.memory` y `performance.timeOrigin`. Estos detalles se obtienen cuando te conectas al navegador.



CARACTERÍSTICAS

- **Objetos de página** : los objetos de página de Watir nos ayudarán a reutilizar el código en forma de clases. Con esta función, podemos automatizar nuestra aplicación sin tener que duplicar ningún código y, además, hacerla más manejable.
- **Descargas** : con Watir, es fácil probar la descarga de archivos para la interfaz de usuario o el sitio web.
- **Alertas** : Watir proporciona API fáciles de usar para probar las alertas emergentes en su interfaz de usuario o sitio web.
- **Pruebas sin interfaz gráfica** : con las pruebas sin interfaz gráfica, los detalles se obtienen en la línea de comandos sin tener que abrir el navegador. Esto ayuda a ejecutar casos de prueba de IU en la línea de comandos.

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

@selection at the end -add
mirror_ob.select= 1
mirror_ob.select=1
context.scene.objects.active
("Selected" + str(modifier))
mirror_ob.select = 0
bpy.context.selected_objects
data.objects[one.name].select
print("please select exactly one object")
- OPERATOR CLASSES -
types.Operator:
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"
context):
    context.active_object is not None
```

VENTAJAS

- Watir es una herramienta de código abierto y muy fácil de utilizar.
- Watir está desarrollado en Ruby y cualquier aplicación web que funcione en un navegador se puede automatizar fácilmente usando Watir.
- Watir es compatible con todos los navegadores más recientes, lo que facilita las pruebas.
- Watir tiene bibliotecas integradas para probar el rendimiento de la página, alertas, probar iframes, ventanas del navegador, tomar capturas de pantalla, etc.

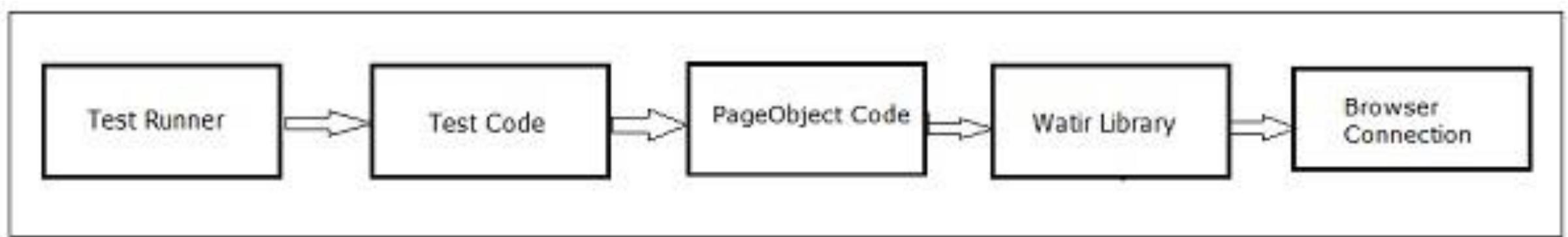




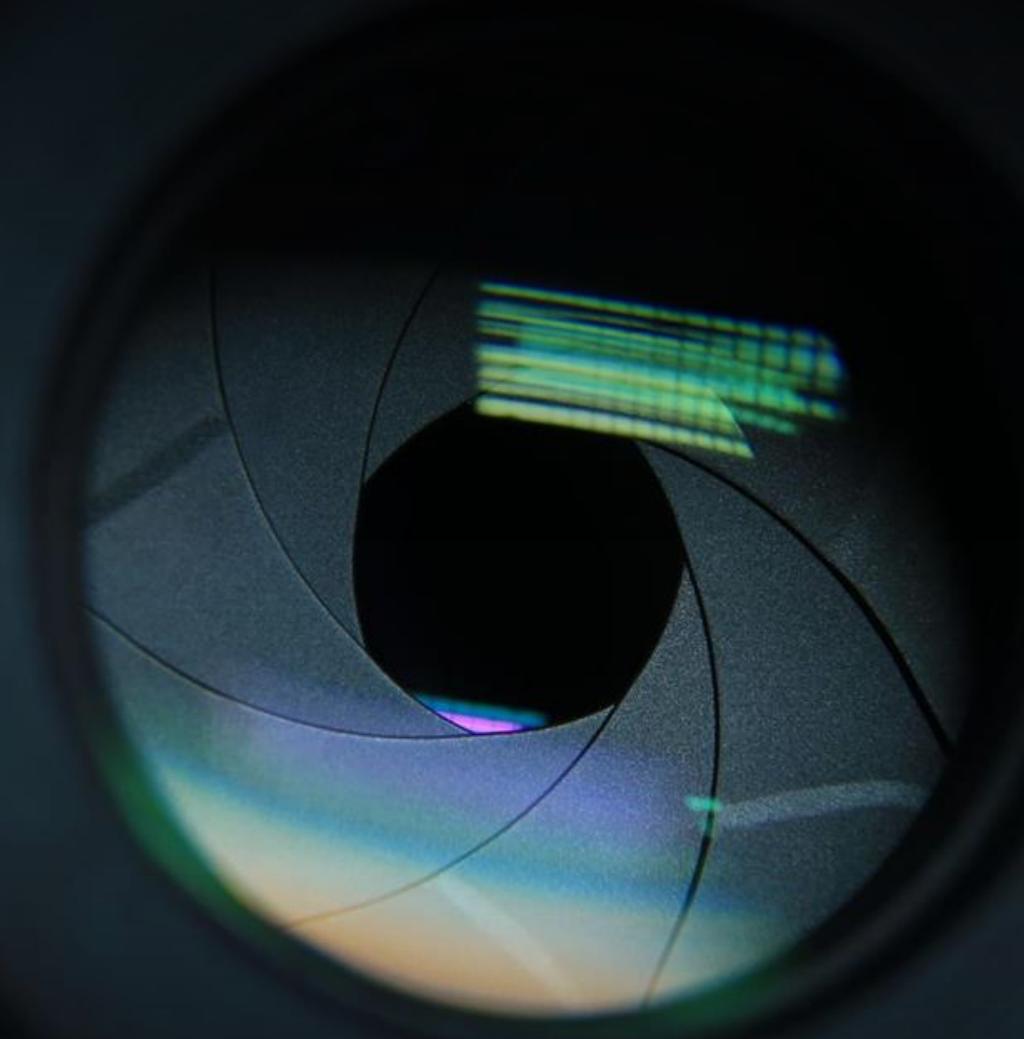
DESVENTAJAS

Watir solo es compatible con el marco de pruebas Ruby y no se puede utilizar con ningún otro marco de pruebas.

Las pruebas móviles con Watir no están mejoradas y los navegadores de escritorio se imitan para comportarse como navegadores móviles en lugar de actuar como dispositivos en tiempo real.



DEMO



EJERCICIO 3

```
▶ packages
  ▶ scripts
  ▶ src
  ▶ test
  ▶ types
  .babelrc.js
  .editorconfig
  .eslintignore
  .eslintrc.js
  .flowconfig
  .gitignore
  BACKERS.md
  LICENSE
```

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more attributes (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template compiler

EJERCICIO

4

▶ packages
▶ scripts
▶ src
▶ test
▶ types

.babelrc.js

.editorconfig

.eslintignore

.eslintrc.js

.flowconfig

.gitignore

BACKERS.md

LICENSE

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

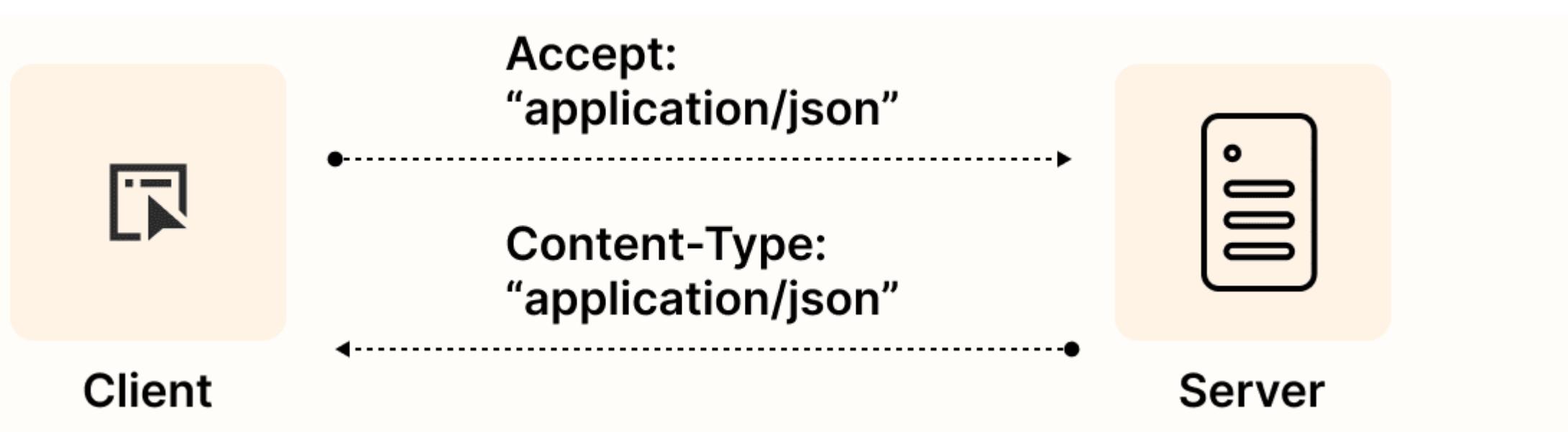
feat: dynamic directive arguments for v-bind and v-on

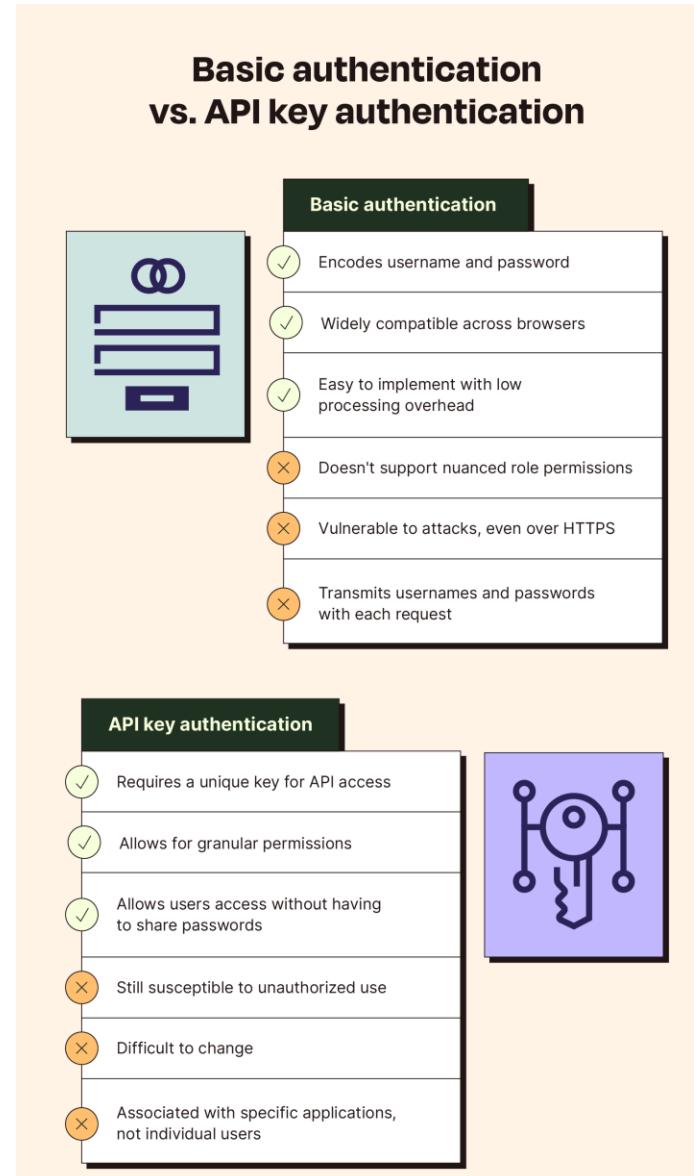
refactor: extend dom-props update skip to more attributes (#9372)

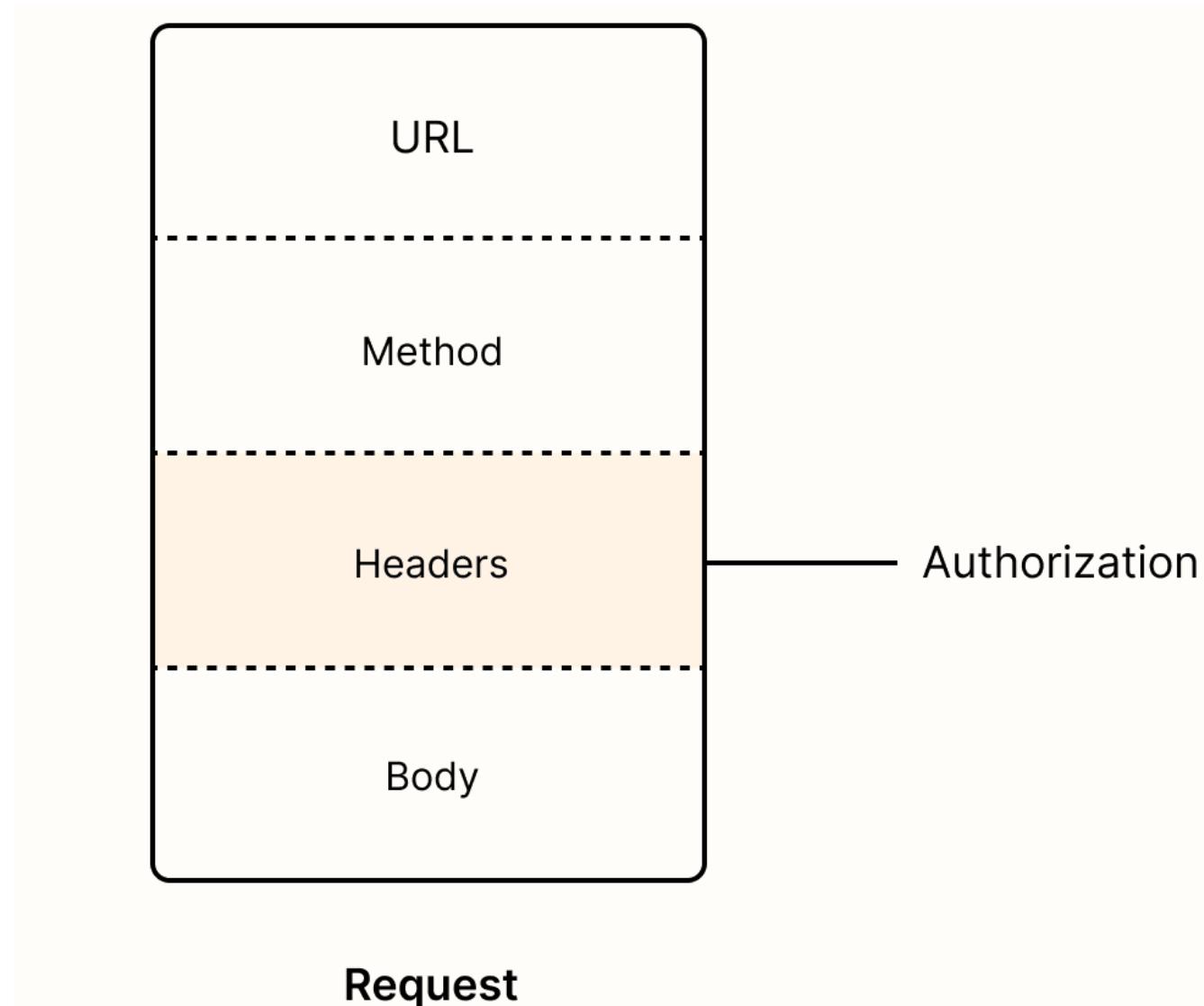
fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template compiler









SMTP & API

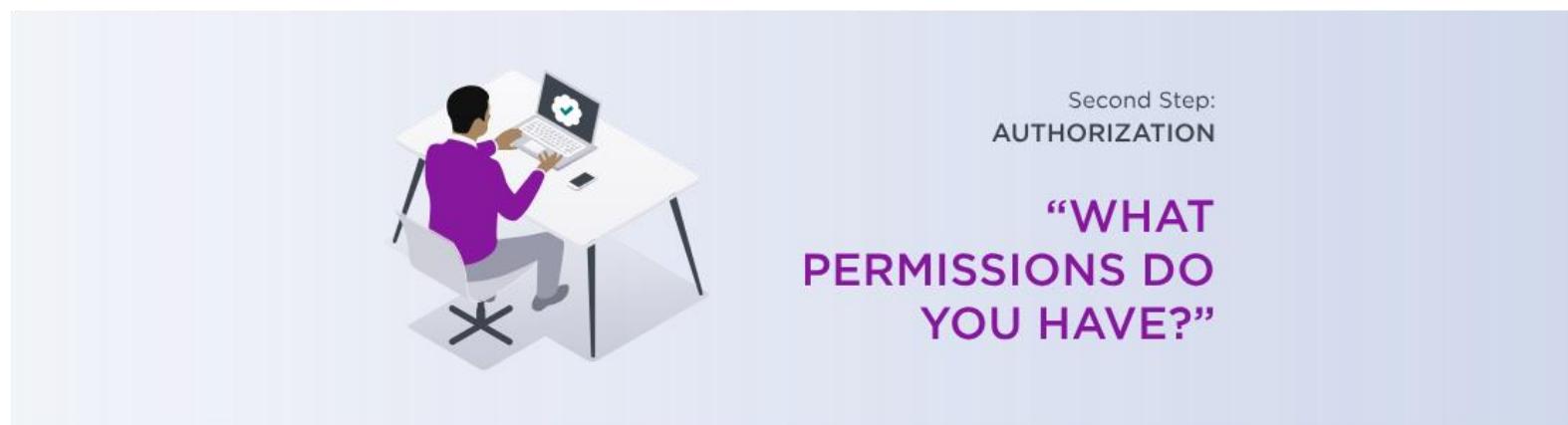
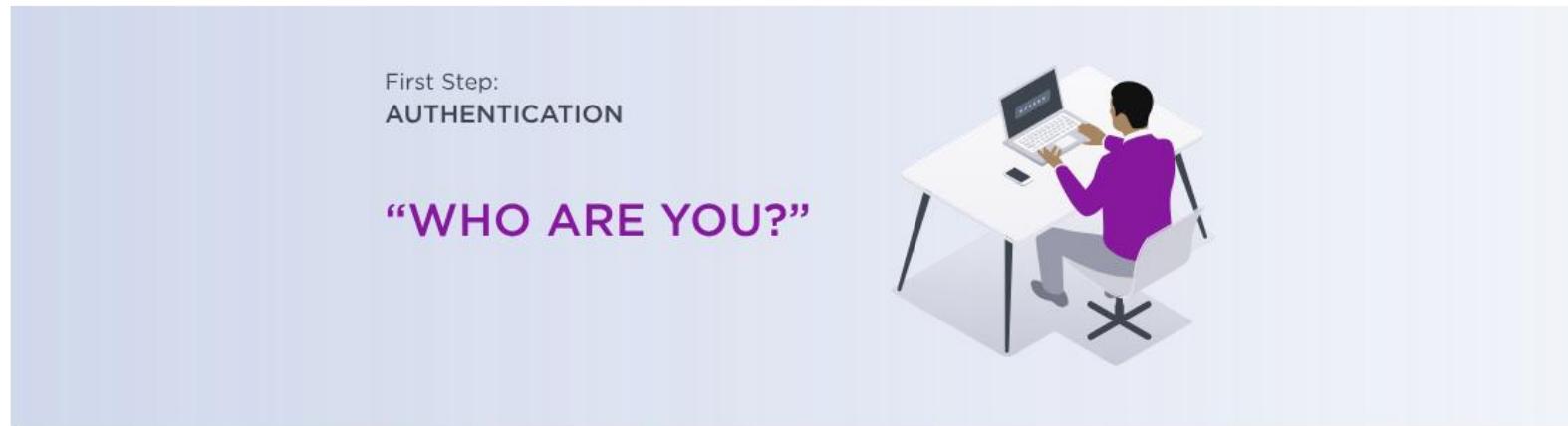
[Generate a new API key](#)[SMTP](#) [API Keys](#)

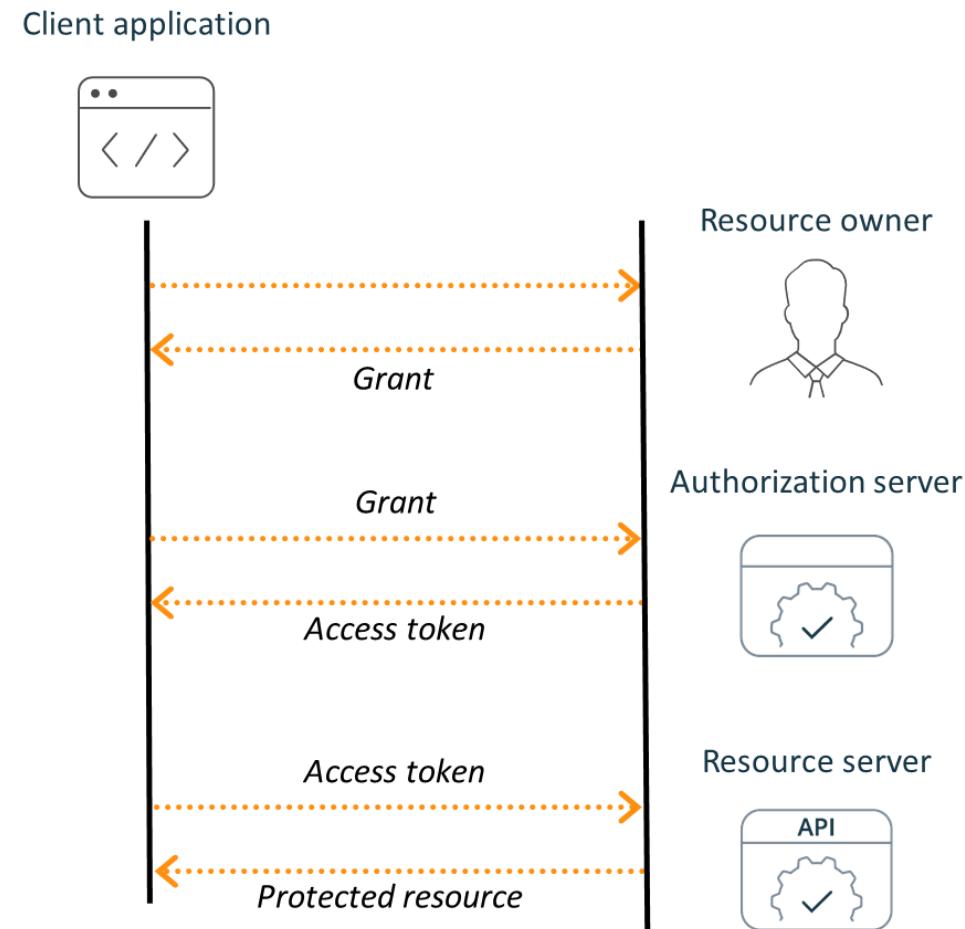
Your API Keys

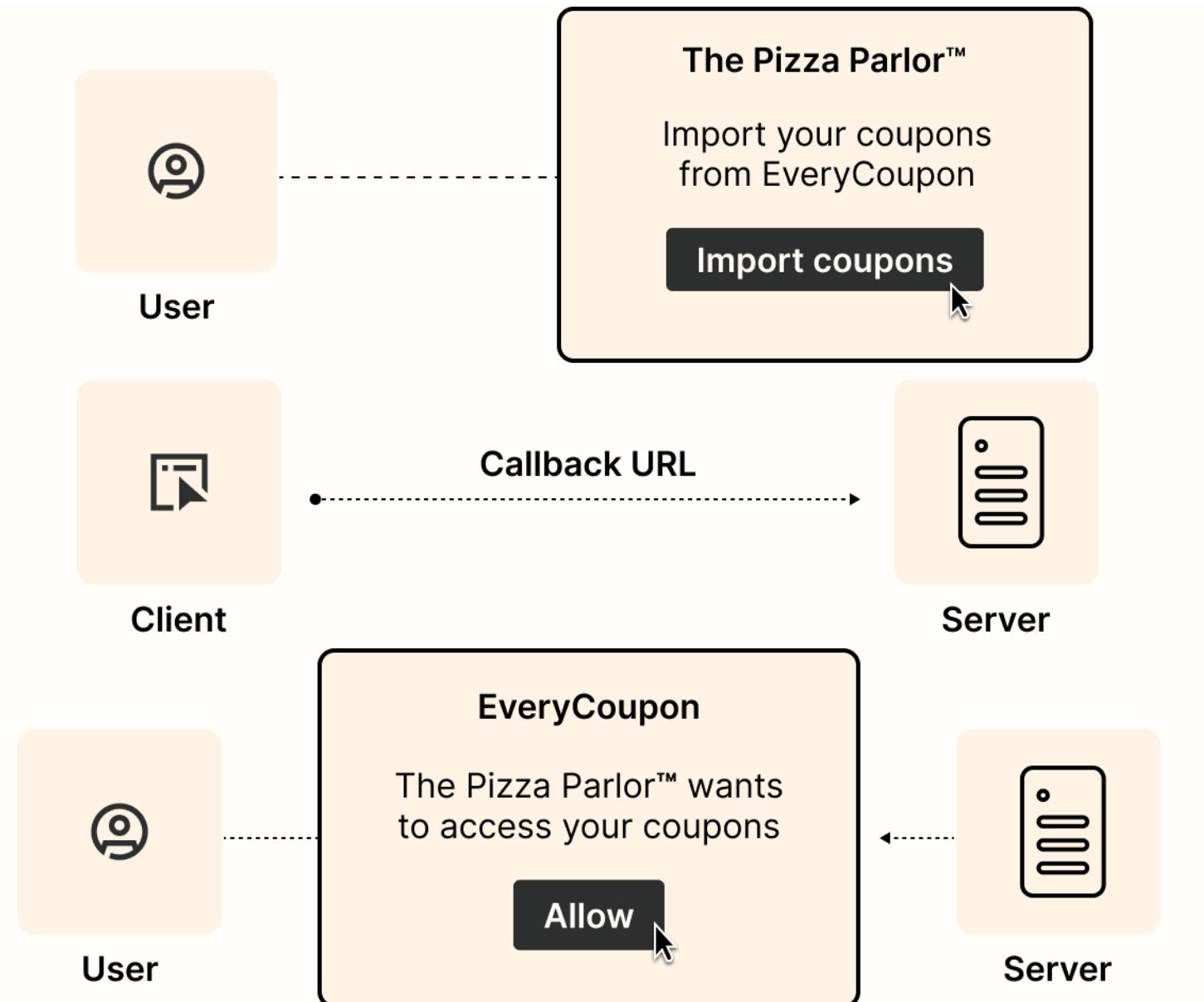
Version	API Key	Name	Created on
<input type="checkbox"/> v3	*****Xo4rJT	prestashop 1.7	April 4, 2023 12:48 PM
<input type="checkbox"/> v3	*****QnbN3k	WordPress	

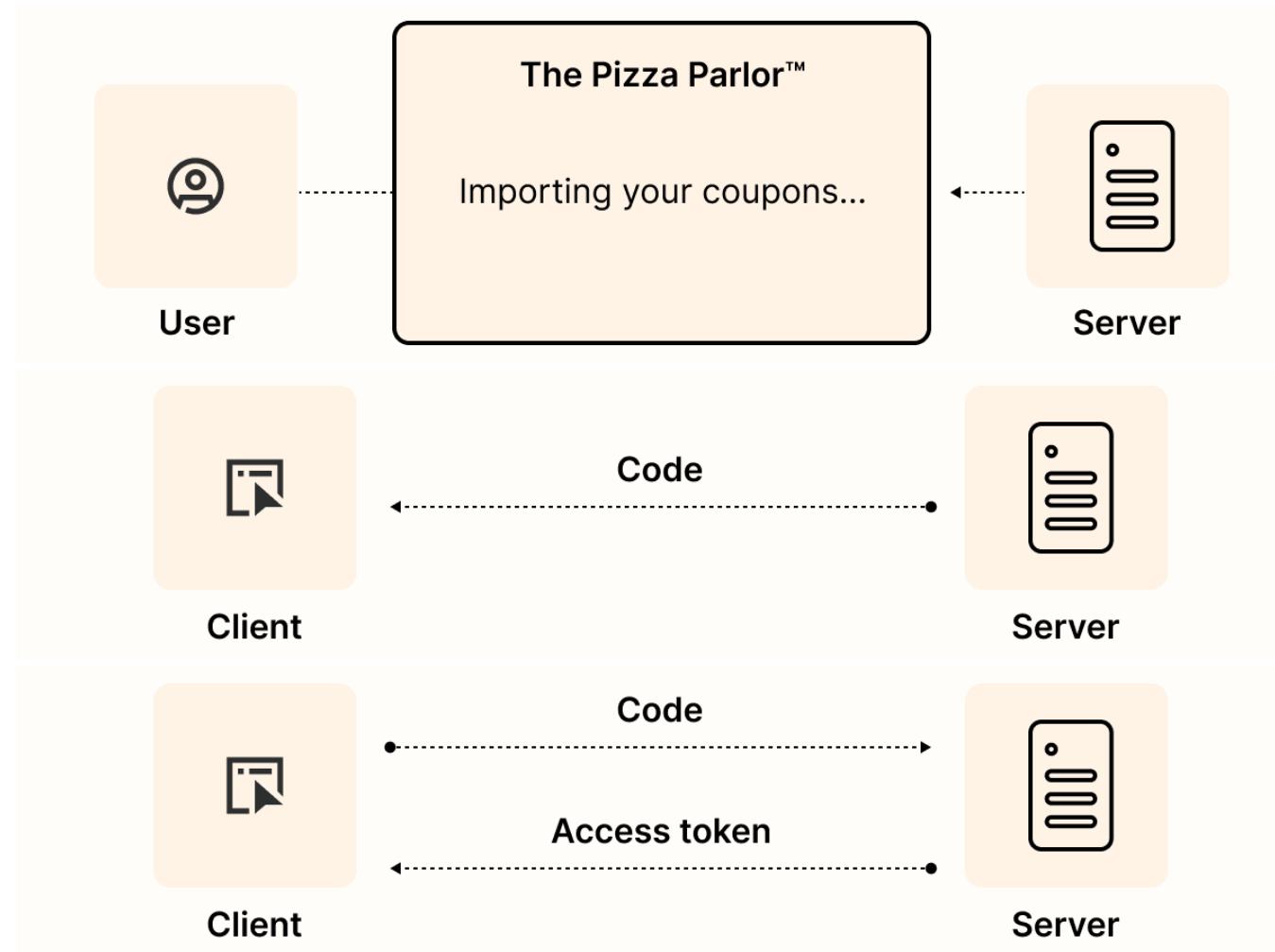


AUTHENTICATION VS. AUTHORIZATION





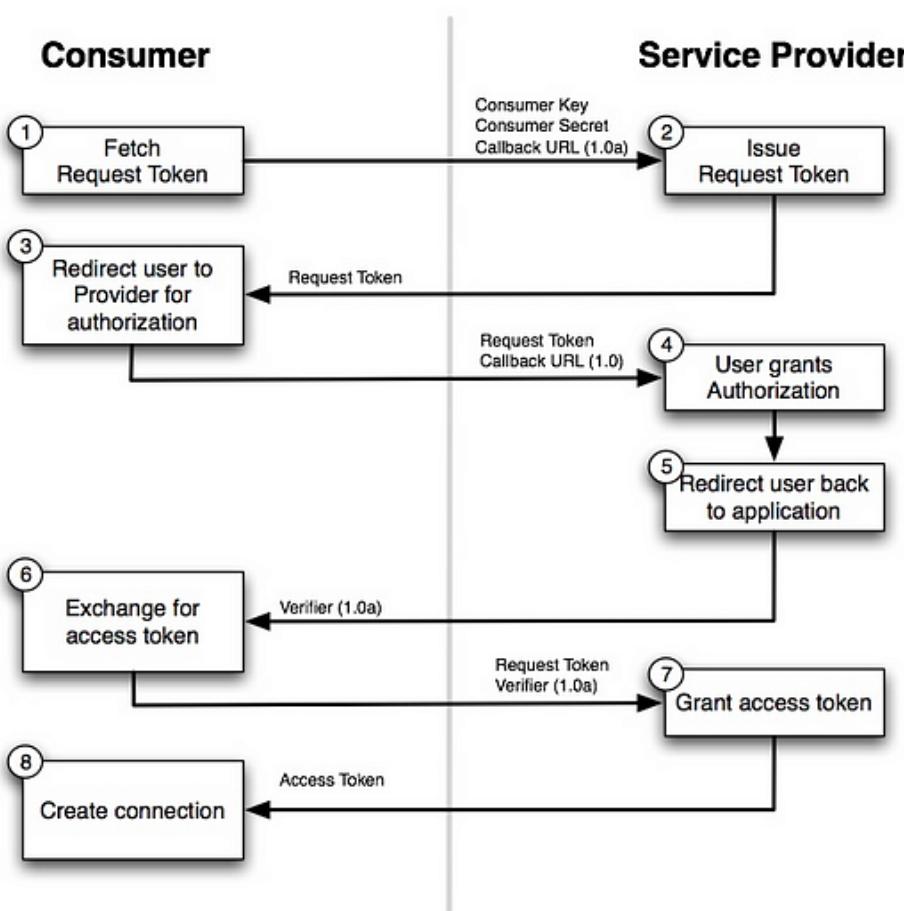




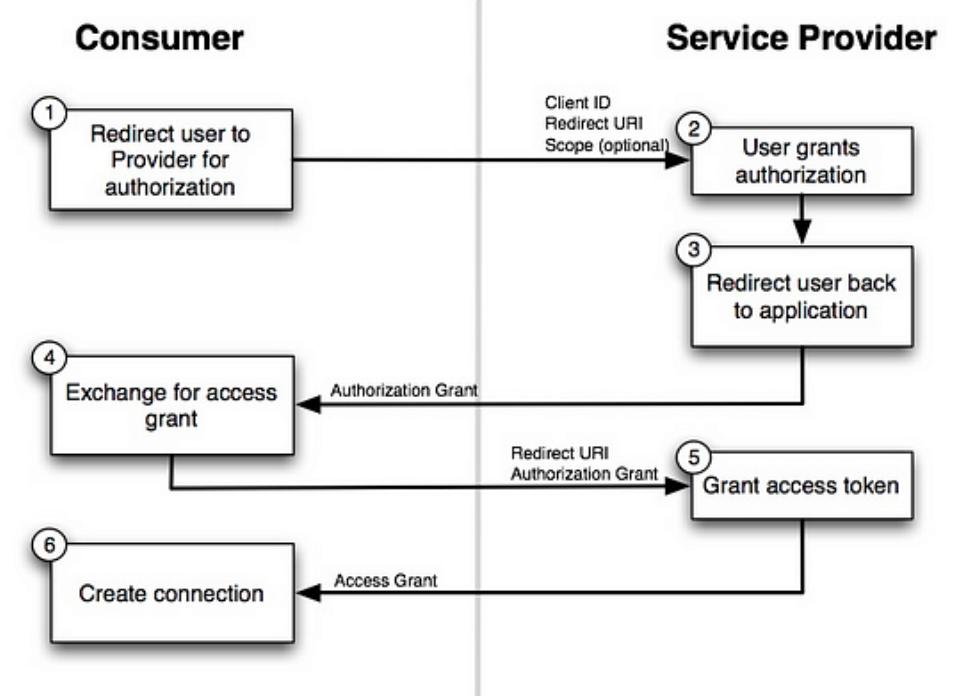


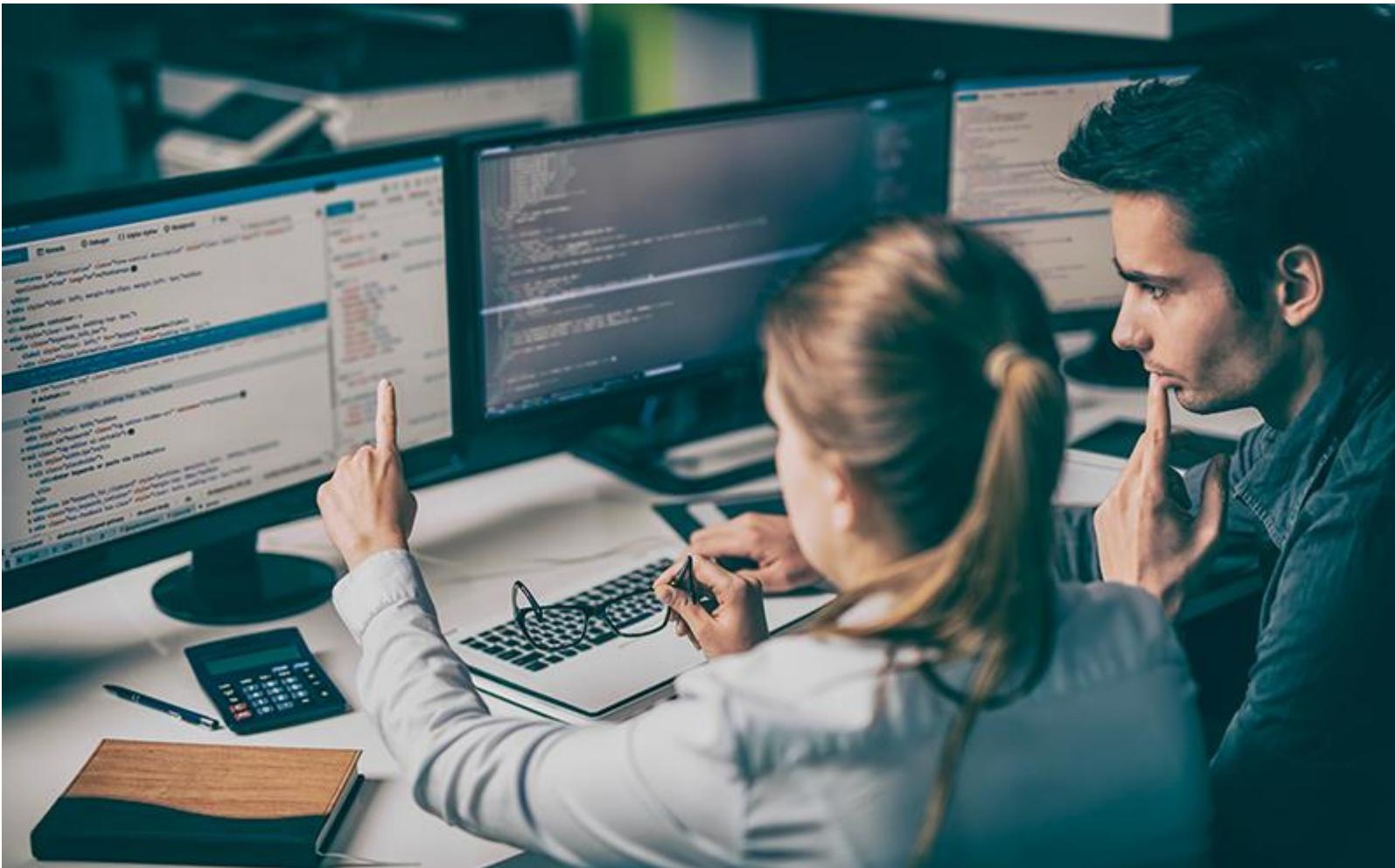


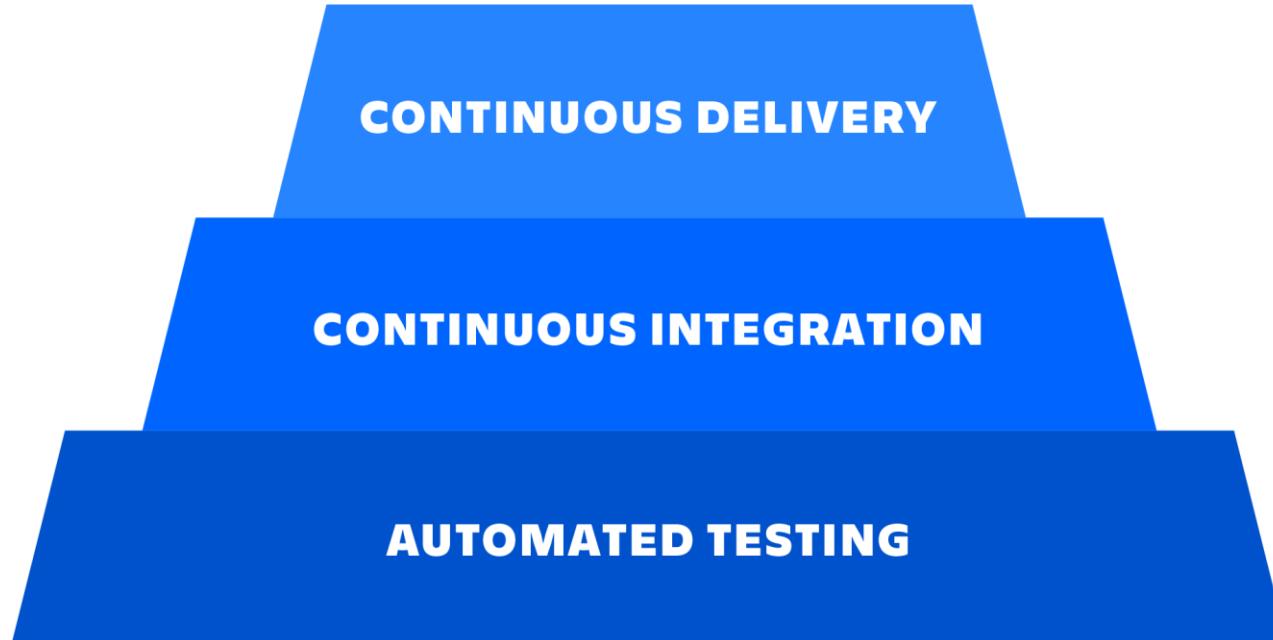
OAuth 1.0

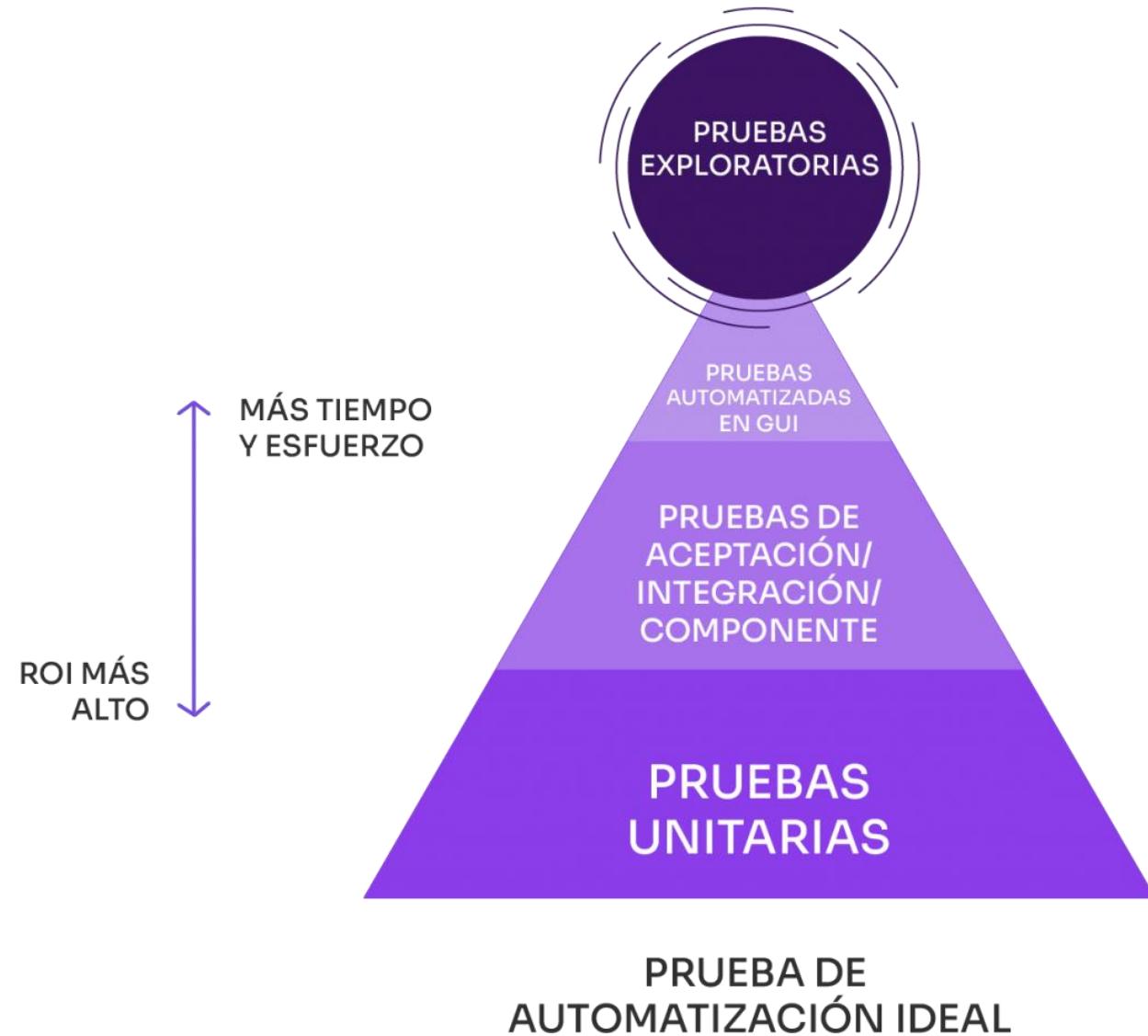


OAuth 2.0



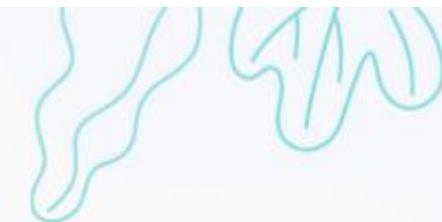








Tipos de Pruebas API



Funcionales



Seguridad



Rendimiento



Integración



Documentación





Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import GET Untitled Request + ... No Environment

Collections APIs Environments Mock Servers Monitors Flows History

Media Authentication API

- POST Authentication
- POST Access Token
- GET User's Page
- GET Instagram Account
- GET New Request

Untitled Request

Enter request URL

Send

Params Authorization Headers (5) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create collection

Response

Enter the URL and click Send to get a response

Bootcamp Auto-select agent Runner Trash ?

Console



History		Collections
All	Me	Team
	Regression_test	...
	1_Incorrect_login_admin	
	2_Correct_login_admin	
	3_Create_employee2	
	4_Get_employee2	
	5_Modify_employee2	
	6_Get_employees	
	7_Delete_employee2	
	8_Get_employee2	
	9_Get_employees	
	10_Create_service1	
	11_Modify_service1	
	12_Get_service1	
	13_Get_services	
	14_Create_client1	
	15_Modify_client1	
	16_Get_client1	
	17_Create_appointment1	



MANAGE ENVIRONMENTS

Manage Environments Environment Templates

Environments help you customize requests according to variables. [Learn More](#)

API-Test	Share	⋮	⋮
pushServer	Share	⋮	⋮
pushServer 3333	Share	⋮	⋮
SYSDULE-ALL-QA	Share	⋮	⋮
SYSDULE-QA	Share	⋮	⋮

Globals Import Add

Figura 3. Gestión de entornos.





The screenshot shows the Postman application interface with various UI elements highlighted by numbered callouts:

- 1**: New button in the top bar.
- 2**: File menu item.
- 3**: Edit menu item.
- 4**: View menu item.
- 5**: Runner button in the top bar.
- 6**: Invite button in the top bar.
- 7**: History button in the left sidebar.
- 8**: Collections button in the left sidebar, which is currently selected.
- 9**: Untitled Request title in the main workspace.
- 10**: GET method dropdown.
- 11**: Send button.
- 12**: Save dropdown.
- 13**: Params tab in the request configuration.
- 14**: Authorization tab in the request configuration.
- 15**: Headers tab in the request configuration.
- 16**: Body tab in the request configuration.
- 17**: Pre-request Script tab in the request configuration.
- 18**: Tests tab in the request configuration.

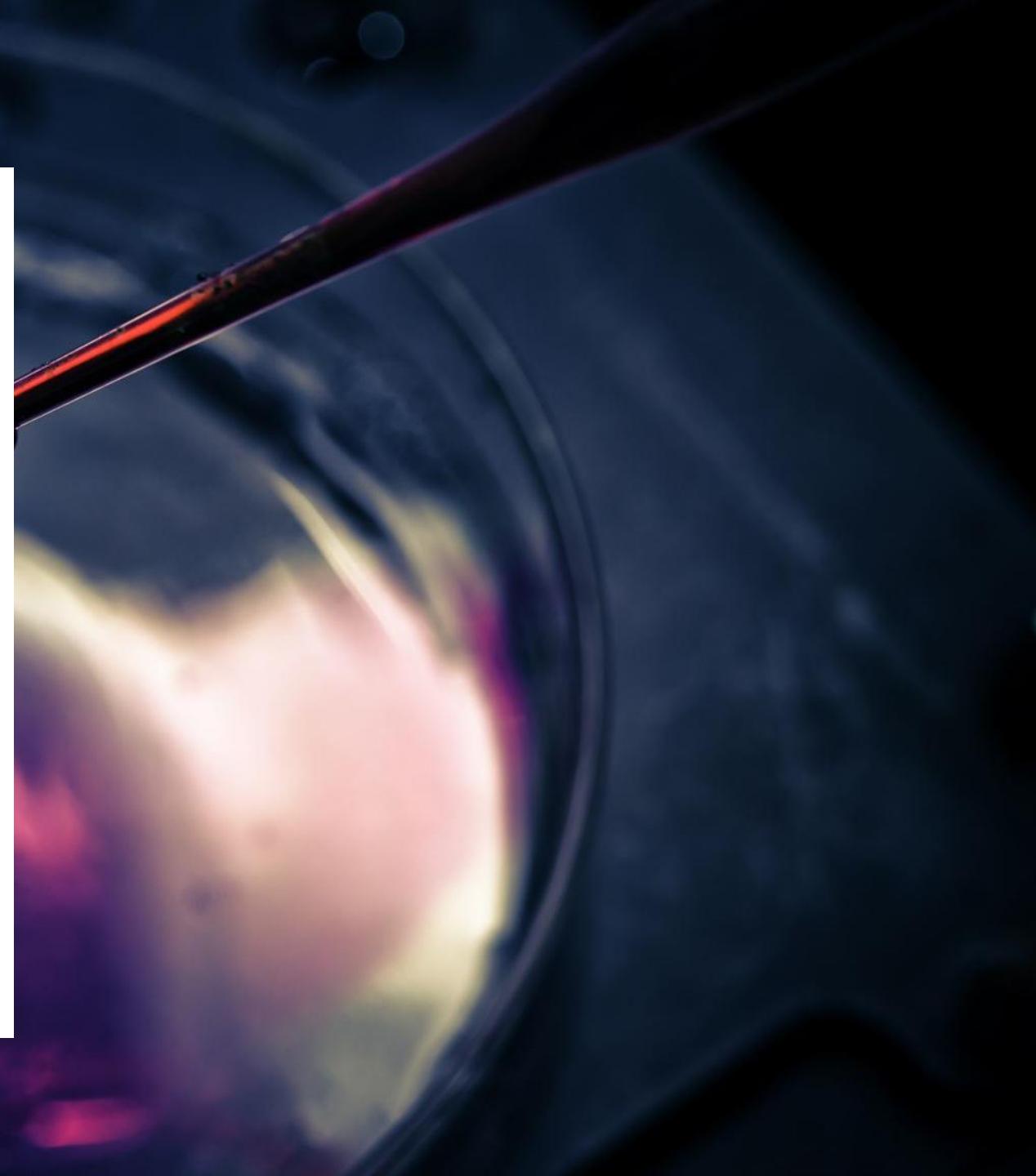
The main workspace displays the following text and interface elements:

- "You don't have any collections"
- "Collections let you group related requests, making them easier to access and run."
- "Create a collection" button
- "Hit the Send button to get a response." text



Soap UI: Elementos Básicos

- Soap UI es una herramienta para pruebas de servicios web y APIs.
- Permite crear y ejecutar pruebas funcionales y de carga.
- Facilita la generación de mocks para pruebas.



PRUEBAS DE SERVICIOS WEB

- **Pruebas de servicios web** es un tipo de prueba de software que valida los servicios web. El propósito de las pruebas de servicios web es verificar la funcionalidad, confiabilidad, rendimiento y seguridad de una API (interfaz de programa de aplicación). Las pruebas de servicios web son similares a las pruebas unitarias en algunos casos. Puede probar un servicio web manualmente o crear su propio código de automatización o utilizar una herramienta de automatización disponible como Postman.

XML

GetQuote

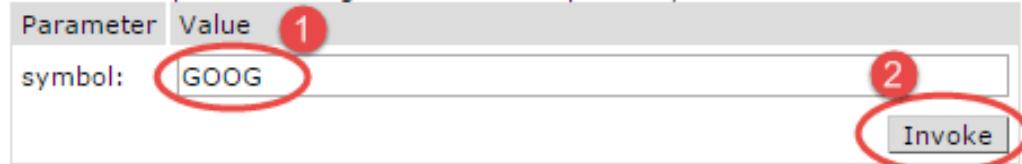
Get Stock quote for a company Symbol

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

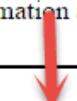
Parameter	Value
symbol:	1 GOOG 2

3 Invoke



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<string xmlns="http://www.webserviceX.NET/">
  <StockQuotes><Stock><Symbol>GOOG</Symbol><Last>534.03</Last><Date>12/26/2014</Date>
  <Time>4:00pm</Time><Change>+5.26</Change><Open>528.77</Open><High>534.25</High>
  <Low>527.31</Low><Volume>1037774</Volume><MktCap>362.3B</MktCap>
  <PreviousClose>528.77</PreviousClose><PercentageChange>+0.99%</PercentageChange>
  <AnnRange>489.00 - 604.83</AnnRange><Earns>19.002</Earns><P-E>27.83</P-E>
  <Name>Google Inc.</Name></Stock></StockQuotes>
</string>
```



PROTOCOLOS

- Los servicios web se pueden implementar de diferentes maneras, pero los dos siguientes son los enfoques de implementación más populares.
- SOAP (Protocolo simple de acceso a objetos)
- REST (Arquitectura de transferencia de estado representacional)



REST

- REST significa REpresentational State Transfer; es una arquitectura que generalmente se ejecuta sobre HTTP. El estilo REST enfatiza las interacciones entre clientes y servicios, que se mejoran al tener un número limitado de operaciones. REST es una alternativa a SOAP (Simple Object Access Protocol) y en lugar de usar XML para las solicitudes, REST usa URL simples en algunos casos. A diferencia de SOAP, las aplicaciones RESTFUL usan encabezados HTTP integrados para transportar metainformación.
-



WSDL

- WSDL (servicios web Description Language) es un lenguaje basado en XML que se utilizará para describir los servicios ofrecidos por un servicio web.
- WSDL describe todas las operaciones que ofrece el servicio web en particular en formato XML. También define cómo se pueden llamar los servicios, es decir, qué valor de entrada debemos proporcionar y cuál será el formato de la respuesta que generará para cada tipo de servicio.

The image shows a close-up of a person's hands typing on a dark-colored computer keyboard. In the background, a computer monitor displays a dark-themed code editor with yellow and white text. The code appears to be Python, related to 3D modeling, specifically concerning object mirroring. Some parts of the code are blurred due to motion or focus.

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
bpy.context.scene.objects.active = ("Selected" + str(modifier))
mirror_ob.select = 0
bpy.context.selected_objects = bpy.data.objects[one.name].select
print("please select exactly one object")
- - - OPERATOR CLASSES - - -
types.Operator:
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"
context):
    context.active_object is not None
```

COMO PROBAR

- Para probar el servicio web, puede
 - Probar manualmente
 - Crea tu propio código de automatización
 - Utilice una herramienta de automatización lista para usar como SoapUI.

```
    mirror_mod = modifier_obj
    # Set mirror object to mirror
    mirror_mod.mirror_object = mirror_obj

    if operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    elif operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    elif operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

    #selection at the end -add
    mirror_ob.select= 1
    modifier_ob.select=1
    bpy.context.scene.objects.active = modifier_obj
    print("Selected" + str(modifier_obj))
    mirror_ob.select = 0
    bpy.context.selected_objects.append(mirror_obj)
    data.objects[one.name].select = 1
    print("Selected" + str(data.objects[one.name]))
    print("please select exactly one object")

- - - - - OPERATOR CLASSES - - - - -
# Operator class for Mirror modifier
class MirrorOperator(bpy.types.Operator):
    bl_idname = "object.mirror"
    bl_label = "X mirror to the selected object.mirror_mirror_x"
    bl_description = "Mirror X"
    bl_options = {'REGISTER', 'UNDO'}

    def execute(self, context):
        if context.object is None or context.object.type != 'MESH':
            self.report({'ERROR'}, "context: active object is not a mesh")
            return {'CANCELLED'}
        if context.active_object is not None and context.active_object.type != 'MESH':
            self.report({'ERROR'}, "context: active object is not a mesh")
            return {'CANCELLED'}
```

PASOS

- Comprender el archivo WSDL
- Determinar las operaciones que proporciona un servicio web en particular
- Determinar el formato de solicitud XML que debemos enviar.
- Determinar el formato XML de respuesta
- Usar una herramienta o escribir código para enviar una solicitud y validar la respuesta

```
mirror_mod = modifier_obj
# Set mirror object to mirror
mirror_mod.mirror_object = selected_object

operation = "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False

operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False

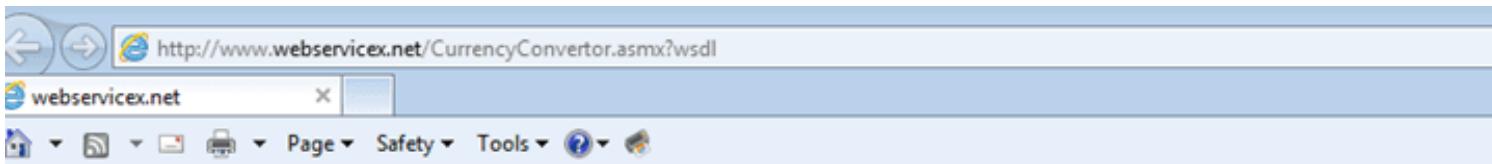
operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

# Selection at the end - add
if ob.select= 1
    mirror_ob.select=1
    context.scene.objects.active = context.scene.objects.get("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects.append(data.objects[one.name])
    print("please select exactly one object")
else:
    print("please select exactly one object")

- OPERATOR CLASSES ---

types.Operator:
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"

context):
    context.active_object is not None
```



```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://www.webserviceX.NET"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.webserviceX.NET/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:sc="http://schemas.xmlsoap.org/wsdl/sequence/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/">
  - <wsdl:types>
    - <s:schema targetNamespace="http://www.webserviceX.NET/" elementFormDefault="qualified">
      - <s:element name="ConversionRate">
          - <s:complexType>
            - <s:sequence>
              <s:element name="FromCurrency" type="tns:Currency" maxOccurs="1" minOccurs="1"/>
              <s:element name="ToCurrency" type="tns:Currency" maxOccurs="1" minOccurs="1"/>
            </s:sequence>
          </s:complexType>
        </s:element>
      - <s:simpleType name="Currency">
          - <s:restriction base="s:string">
              <s:enumeration value="AFA"/>
              <s:enumeration value="ALL"/>
              <s:enumeration value="DZD"/>
              <s:enumeration value="ARS"/>
              <s:enumeration value="AWG"/>
              <s:enumeration value="AUD"/>
              <s:enumeration value="BSD"/>
              <s:enumeration value="BHD"/>
              <s:enumeration value="BDT"/>
              <s:enumeration value="BBD"/>
              <s:enumeration value="BZD"/>
              <s:enumeration value="BMD"/>
              <s:enumeration value="BTN"/>
              <s:enumeration value="BOB"/>
              <s:enumeration value="BWP"/>
              <s:enumeration value="BRL"/>
```

WebService method which
we will use to get the
conversion rate between 2
currencies

Currency name abbreviations for different
countries to be used while conversions

`<s:enumeration value="TRY"/>
 </s:restriction>
 </s:simpleType>
- <s:element name="ConversionRateResponse">
 - <s:complexType>
 - <s:sequence>
 <s:element name="ConversionRateResult" type="s:double" maxOccurs="1" minOccurs="1"/>
 </s:sequence>
 </s:complexType>
- <s:element name="double" type="s:double"/>
 </s:schema>
- <wsdl:types>
- <wsdl:message name="ConversionRateSoapIn">
 <wsdl:part name="parameters" element="tns:ConversionRate"/>
- <wsdl:message name="ConversionRateSoapOut">
 <wsdl:part name="parameters" element="tns:ConversionRateResponse"/>
- <wsdl:message name="ConversionRateHttpGetIn">
 <wsdl:part name="FromCurrency" type="s:string"/>
 <wsdl:part name="ToCurrency" type="s:string"/>
- <wsdl:message name="ConversionRateHttpGetOut">
 <wsdl:part name="Body" element="tns:double"/>
- <wsdl:message name="ConversionRateHttpPostIn">
 <wsdl:part name="FromCurrency" type="s:string"/>
 <wsdl:part name="ToCurrency" type="s:string"/>
- <wsdl:message name="ConversionRateHttpPostOut">
 <wsdl:part name="Body" element="tns:double"/>
- <wsdl:portType name="CurrencyConvertorSoap">
 - <wsdl:operation name="ConversionRate">
 <wsdl/documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
Get conversion rate from
face='Verdana'><u>Different currency Code and Names around the world</u></p>'
Albanian Lek
DZD-Algerian Dinar
ARS-Argentine Peso
AWG-Aruba Florin
AUD-Aust
Taka
BBD-Barbados Dollar
BZD-Belize Dollar
BMD-Bermuda Dollar
BTN-Bhutan Ngul
Real
GBP-British Pound
BND-Brunel Dollar
BIF-Burundi Franc
XOF-CFA Franc (BCE
Dollar
CVE-Cape Verde Escudo
KYD-Cavman Islands Dollar
CLP-Chilean Peso
CNY-(`

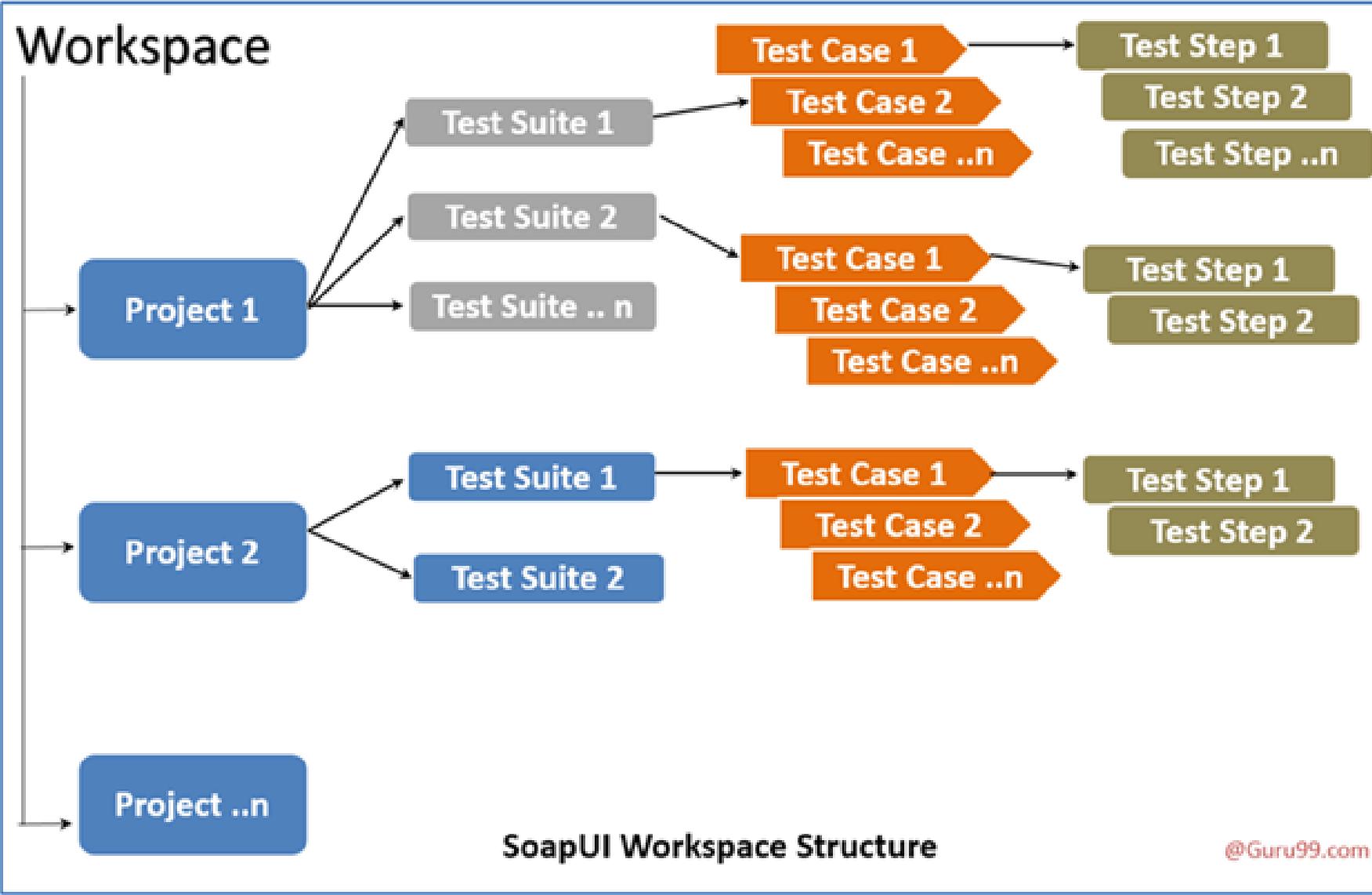
Webservice response that will give us the conversion rate

HTML get and post request details.
We can use any one to send request along with parameters

CONFIGURACION

- El primer paso en SOAP UI es crear un espacio de trabajo. Puede haber varios proyectos asociados a un espacio de trabajo. Los usuarios pueden crear más de un espacio de trabajo. En la versión SOAP UI Pro, podemos cambiar de entorno sin problemas para asignarlo a diferentes puntos finales.
- Para cada proyecto, podemos crear varios conjuntos de pruebas.
- Para cada conjunto de pruebas, puede haber varios casos de prueba adjuntos.
- Para cada caso de prueba, puede haber varios pasos de prueba asociados.





ENVELOPE (Mandatory): Defines the Beginning and the end of the Message

HEADER - (Optional) - used in processing the message, either at an intermediary point or at the ultimate end point.

SOAP Message

BODY (Mandatory) : Actual message of the XML

FAULT (Optional) - This element that provides information about errors that occurred while processing the message

LABORATORIO 7

```
▶ packages
  ▶ scripts
  ▶ src
  ▶ test
  ▶ types
  .babelrc.js
  .editorconfig
  .eslintignore
  .eslintrc.js
  .flowconfig
  .gitignore
  BACKERS.md
  LICENSE
```

v2.6.0-beta.2 build: release 2.6.0-beta.2

build: build 2.6.0-beta.2

feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)

origin/dynamic-directive-arguments

feat: dynamic args for custom directives

perf: improve scoped slots change detection accuracy (#9371)

test: test cases for v-on/v-bind dynamic arguments

refactor: v-bind dynamic arguments use bind helper

test: fix tests, resolve helper conflict

fix: fix middle modifier

feat: handle dynamic argument for v-bind.sync

origin/slot-optimization

perf: improve scoped slots change detection accuracy (#9371)

feat: dynamic directive arguments for v-bind and v-on

refactor: extend dom-props update skip to more attributes (#9372)

fix: fix checkbox event edge case in Firefox

test: fix tests in IE/Edge

refactor: simplify template compiler



Property Content



Compliance Status Standard



Script



SLA



JMS



Security

LABORATORIO 8

```
▶ README.md  
▶ .flowconfig  
▶ .gitignore  
▶ BACKERS.md  
▶ LICENSE  
▶ .babelrc.js  
▶ .editorconfig  
▶ .eslintignore  
▶ .eslintrc.js  
▶ .flowconfig  
▶ packages  
▶ scripts  
▶ src  
▶ test  
▶ types
```

v2.6.0-beta.2 build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
feat: dynamic directive arguments for v-on, v-bind and custom directives (#9370)
perf: improve scoped slots change detection accuracy (#9371)
test: test cases for v-on/v-bind dynamic arguments
refactor: v-bind dynamic arguments use bind helper
test: fix tests, resolve helper conflict
fix: fix middle modifier
feat: handle dynamic argument for v-bind.sync
perf: improve scoped slots change detection accuracy (#9371)
feat: dynamic directive arguments for v-bind and v-on
refactor: extend dom-props update skip to more attributes (#9372)
fix: fix checkbox event edge case in Firefox
test: fix tests in IE/Edge
refactor: simplify template compiler (#9373)

BadBoy: Introducción

- BadBoy es una herramienta para pruebas de aplicaciones web.
- Proporciona grabación y reproducción de interacciones.
- Permite la identificación de vulnerabilidades en aplicaciones.

