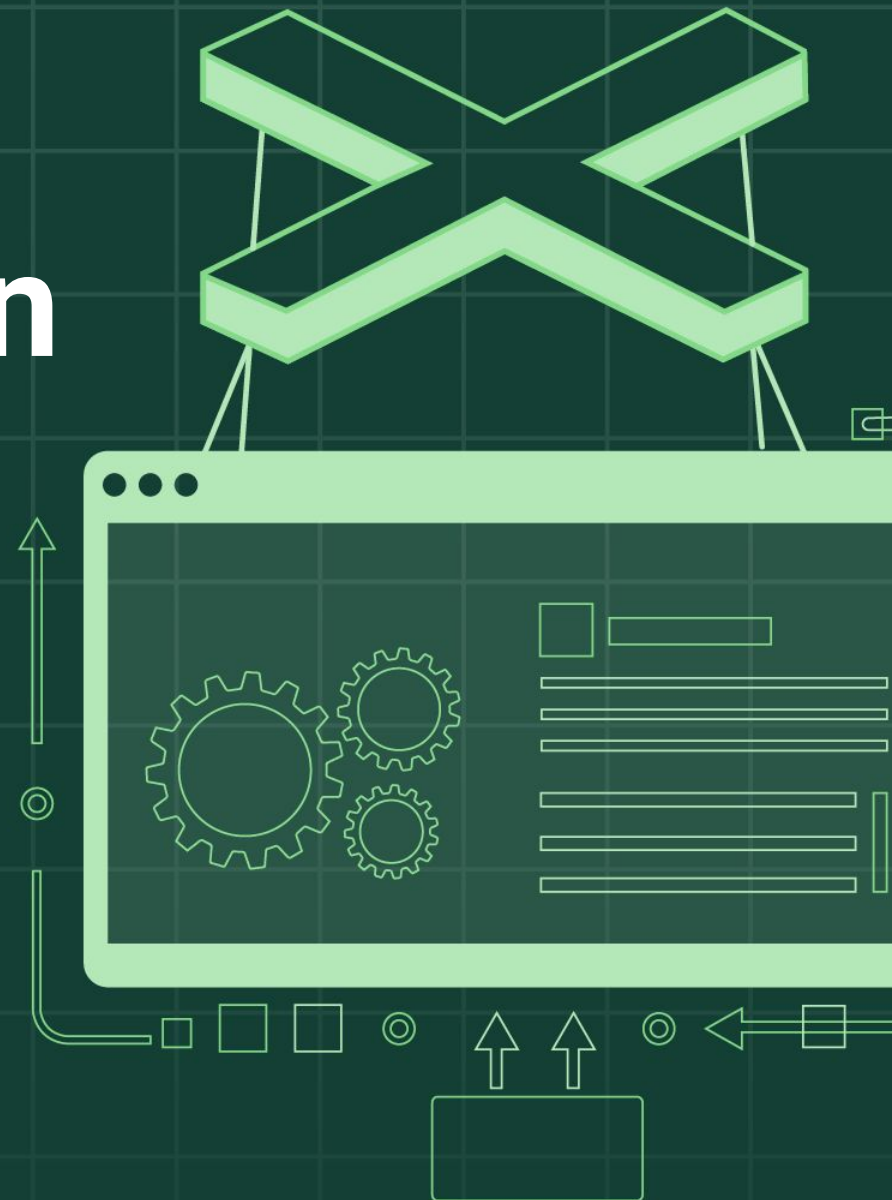


Curso de
**Automatización
de pruebas
con Puppeteer**

Javier Fuentes Mora



■ **Curso de Automatización de**

■ **Pruebas con Puppeteer**

■

Conocimientos previos:



**Curso de Introducción a la
Automatización de Pruebas**

- Javascript
- Node.js

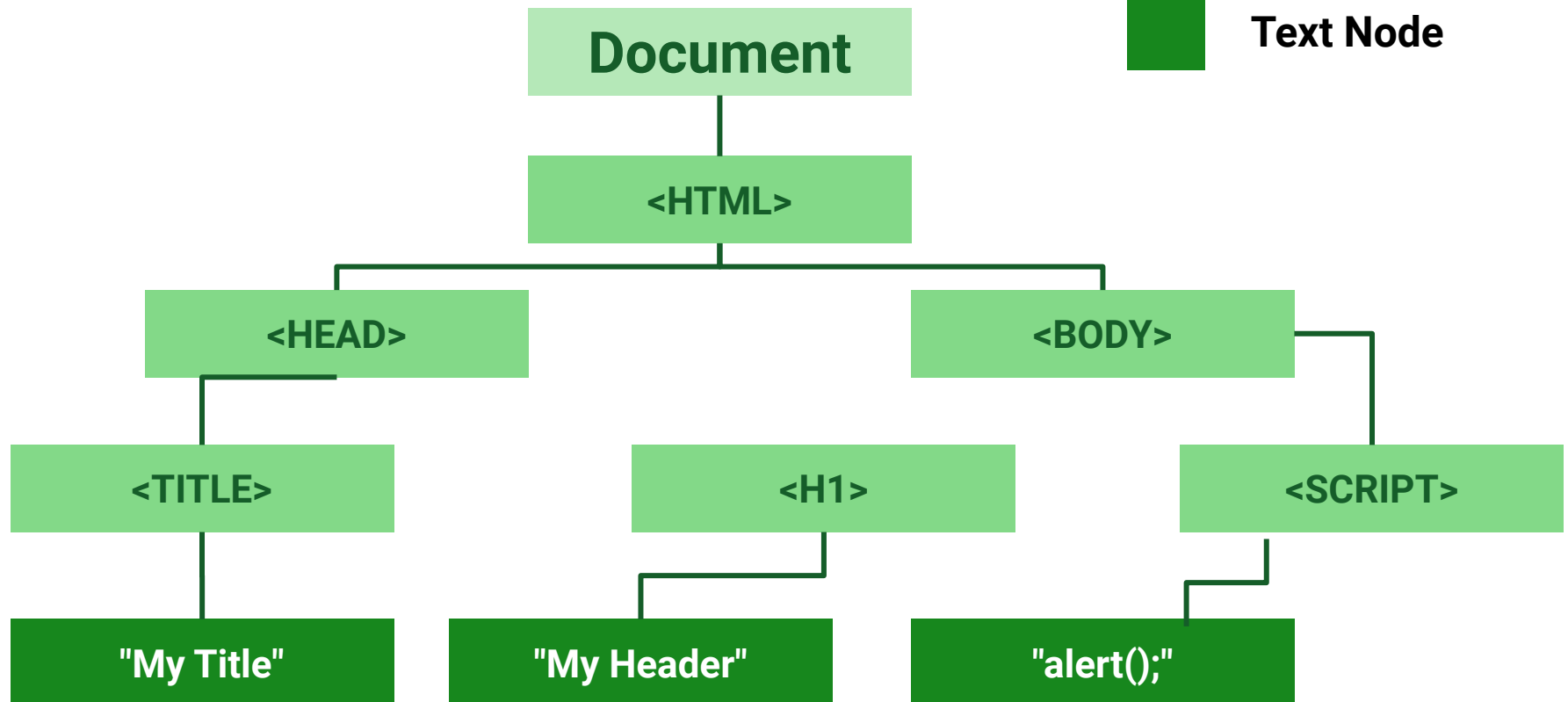
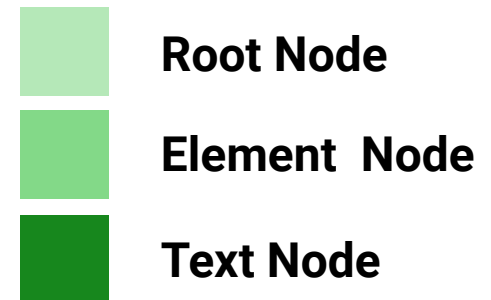
Conocimiento deseado:

- Jest



¿Qué es el **DOM**?

¿Qué es el DOM?



Atributos



Selectores





CSS selectors

- **Universal:** *
- **Tipo:** input
- **Clase:** .classname
- **Atributo:** [attr] [attr=value] [attr~=value]
[attr|=value] [attr^=value] [attr\$=value]
[attr*=value]



CSS selectors

- **Grouping selectors: div, span**
- **Combinadores:**
 - Combinador de hermanos adyacentes: `h1 + p`.
 - Combinador general de hermanos: `p ~ span`.
 - Combinador de hijo: `ul > li`.
 - Combinador de descendientes: `div span`.



CSS selectors

- Pseudo classes: *a : link*.
- Pseudo elementos: *p::first-line*.



Xpath selector

Expresión	Ejemplo
<i>nodename</i>	table
/	/table
//	//table
.	//a[.='Ask Questionmore text']
..	//ul/li/..
@	//@name



Xpath funciones

Función
name()
text()
position()
count()
contains(), starts-with, ends-with
normalize-space()
string-length()
string(), number(), boolean()



Xpath Predicados

Expresión
<code>//div[@class="head"]</code>
<code>//a[@price > 25]</code>
<code>//div[@id="head" and position()=2]</code>
<code>//a[1]</code>
<code>/a[last()]</code>
<code>//button[text()="Submit"]</code>
<code>//section[.//h1[@id='hi']]</code>



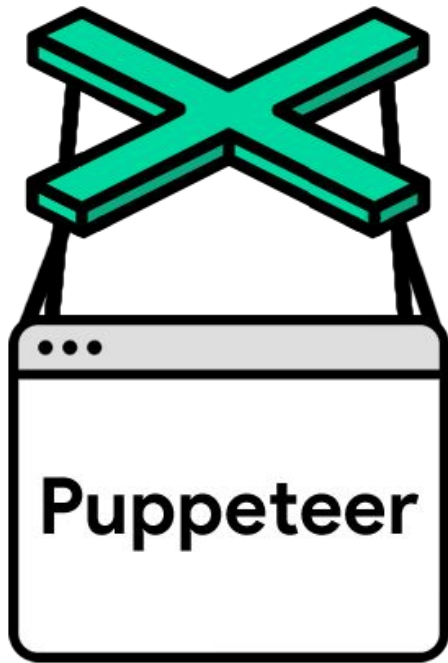
CSS VS Xpath

1	Goal	CSS 3	XPath
2	All Elements	*	//*
3	All P Elements	p	//p
4	All Child Elements	p>*	//p/*
5	Element By ID	#foo	//*[@id='foo']
6	Element By Class	.foo	//*[contains(@class,'foo')]
7	Element With Attribute	*[title]	//*[@title]
8	First Child of All P	p>*:first-child	//p/*[0]
9	All P with an A child	Not possible	//p[a]
10	Next Element	p + *	//p/following-sibling::*[0]
11	Previous Element	Not possible	//p/preceding-sibling::*[0]



¿Qué es Puppeteer?

Puppeteer vs. Selenium





Pros

- **Ofrece más control sobre el navegador.**
- Elimina la dependencia de un driver externo.
- Comparación de imágenes nativas.
- Modo Headless por defecto y mucho más rápido.
- Permite web scraping.
- Mide los tiempos de renderizado y carga mediante la herramienta de análisis de rendimiento de Chrome.



Contras

- **Solo ofrece soporte con Chrome y con Firefox experimental.**
- Solo soporta Node JS.
- No cuenta con un Grabador.
- Solo soporta automatización Web.



¿Entonces cuál tienes
que elegir?





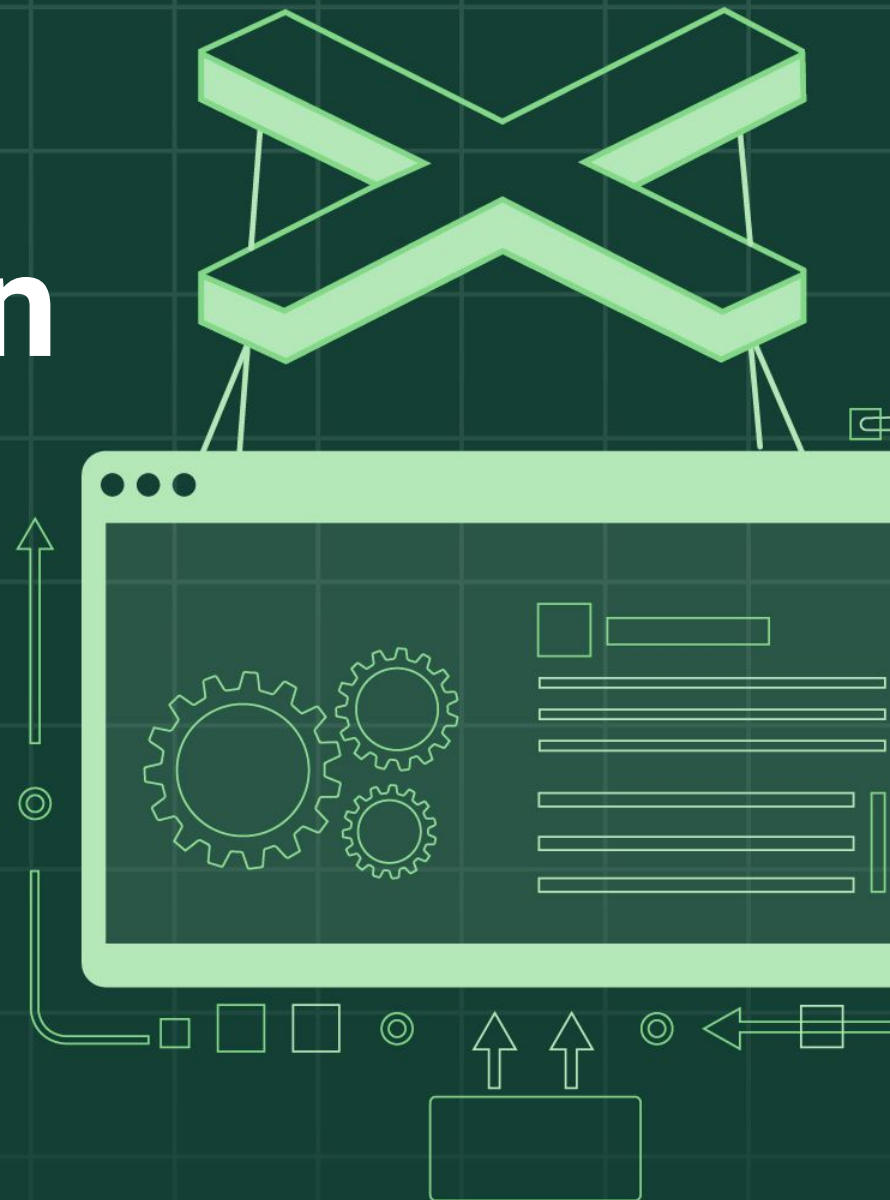
Setup del ambiente

Curso de **Automatización de pruebas con Puppeteer**

Javier Fuentes Mora

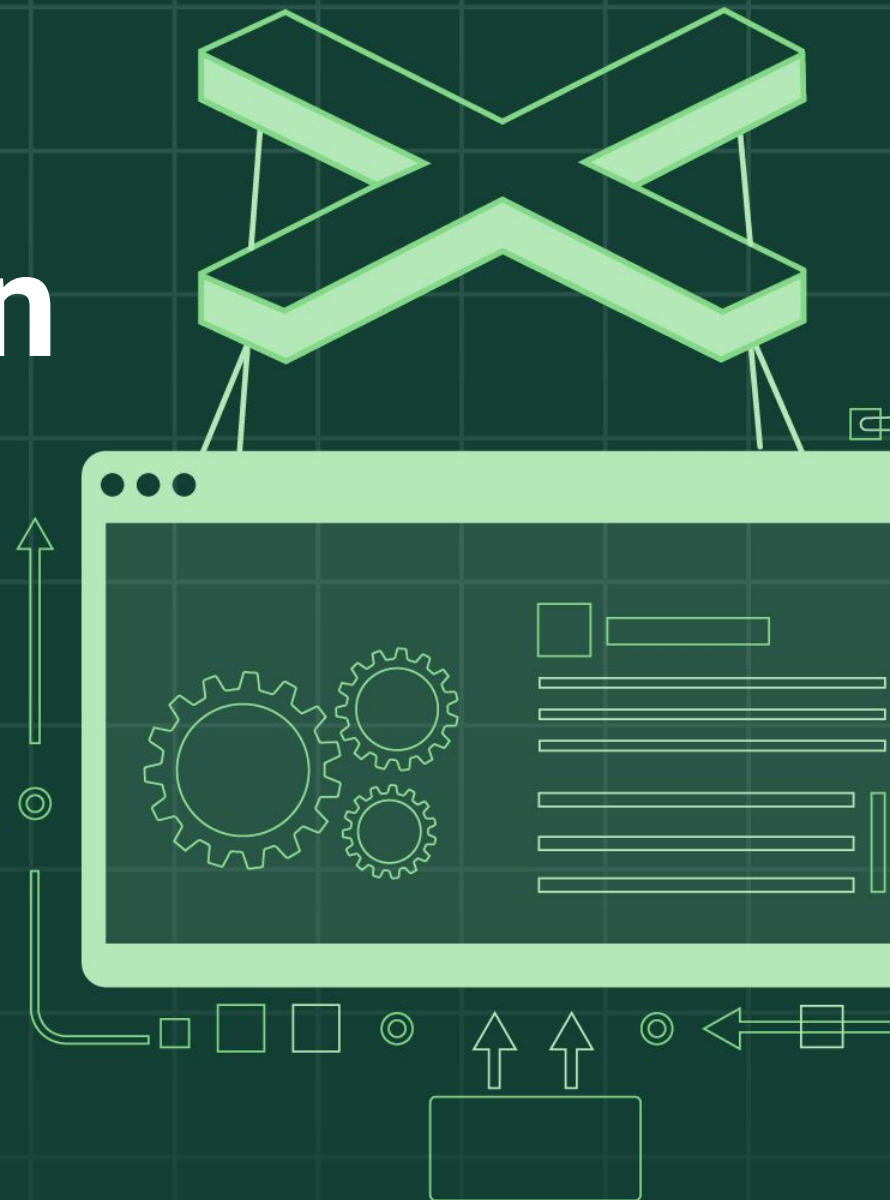
 /javier-fuentes-mora

 @javier_mora94



Curso Avanzado de
**Automatización
de Pruebas
con Puppeteer**

Javier Fuentes Mora



■ **Curso Avanzado de**

■ **Automatización de**

■ **Pruebas con Puppeteer**

Conocimientos previos:

**Curso de Automatización de
Pruebas con Puppeteer**

- Javascript
- Node.js

Conocimientos deseados:

- Jest
- P00
- Jenkins



Setup framework



BDD

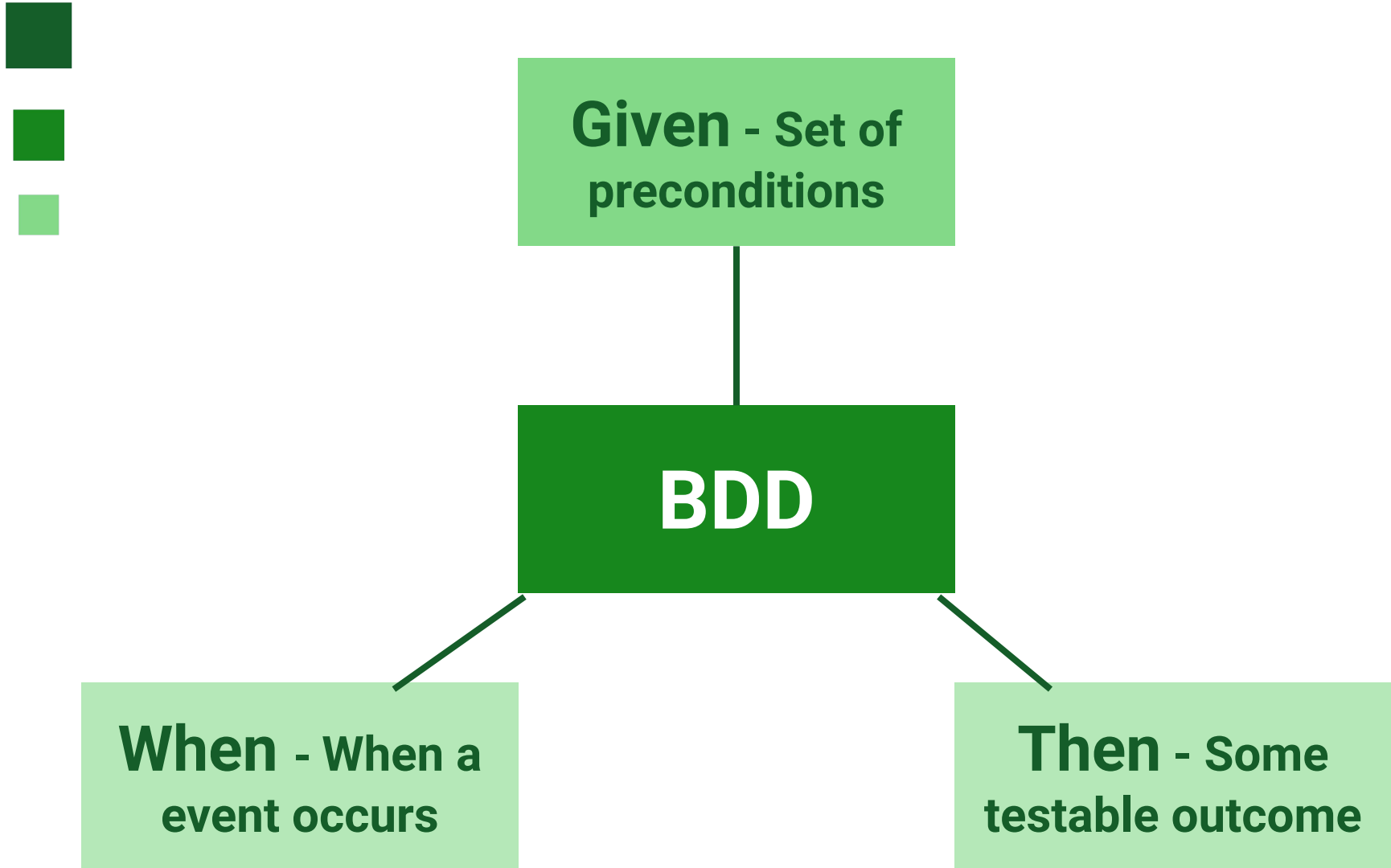
Behavior Driven Development



GHERKIN



GIVEN-WHEN-THEN





Syntaxis Gherkin

Feature: Guess the word

The first example has two steps

Scenario: Maker starts a game

When the Maker starts a game

Then the Maker waits for a Breaker to join

The second example has three steps

Scenario: Breaker joins a game

Given the Maker has started a game with the word "silky"

When the Breaker joins the Maker's game

Then the Breaker must guess a word with 5 characters



Feature

La característica es un texto conciso, pero descriptivo de lo que se desea.

- Se utiliza para describir una función de software y para agrupar diferentes escenarios.



Scenario

Un escenario es un ejemplo o ejemplos que ilustran alguna situación empresarial determinable.

Consiste en una lista de pasos.

- Los escenarios siguen el mismo patrón:
 - Describe un contexto inicial.
 - Describe un evento.
 - Describe un resultado esperado.



Given

El propósito de los GIVEN es describir un estado del sistema conocido antes o precondiciones.

Ejemplos:

- Describir el estado inicial de la base de datos.
- Describir el estado inicial de un usuario (p. Ej.: registrado).



When

El propósito de los WHEN es describir la acción clave que el usuario realiza que desencadena la transición del estado.

Ejemplos:

- Dar clic a un botón.
- Hacer una petición a un endpoint.



Then

El propósito de los THEN es observar los resultados, estas observaciones tienen que estar relacionadas con el valor/beneficio en tu Feature.



Then

Verificar que la combinación de: **GIVEN + WHEN = THEN.**

Ejemplos:

- Ver el Pop Up resultado de una operación.
- Verificar que un sistema externo haya recibido un mensaje esperado.

Ejemplo

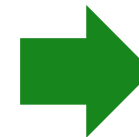
Feature: *Subscribers see different sets of stock images based on their subscription level.*

Scenario: *Free subscribers see only the free articles*

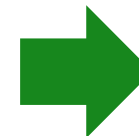
Given *Free Frieda has a free subscription*

When *Free Frieda logs in with her valid credentials.*

Then *she sees a Free Article*



Precondiciones



Disparador



Resultado

Curso Avanzado de **Automatización de Pruebas con Puppeteer**

Javier Fuentes Mora

 /javier-fuentes-mora

 @javier_mora94

