

# WEB SCRAPING Y LAS BASES DE DATOS OFUSCADAS

Abril 2024

Capgemini

# QUÉ APRENDERÁS

---

---

Mundo de los datos

Datos ofuscados

Web scraping

- Normal
- Headless

Selenium

“Defensa” y riesgos del Web scraping

- Algunas estrategias

Testing enfocado a web scraping



# ¿QUIÉN SOY?

---

Pepe Fabra Valverde

Líder y Arquitecto de Front

Haciendo web scraping desde mis inicios, a nivel personal, académico y ligeramente profesional

# AVISO PARA NAVEGANTES

---

- Los recursos se compartirán al final de la sesión
- La sesión **se grabará**, aparecerá el enlace cuando termine
- Es parte de una serie de charlas acerca de Web Scraping



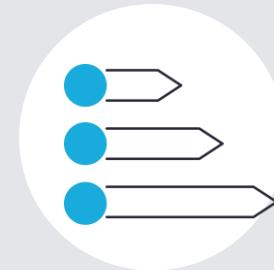
1

## BASES DE DATOS OFUSCADAS

---

# BASES DE DATOS OFUSCADAS

---



**1**

## QUÉ ES UNA BDD

Conceptos esenciales

**2**

## MUNDO DE DATOS

Y por qué importa

**3**

## DATOS OFUSCADOS

Qué son y por qué nos  
importan

**4**

## PIRÁMIDE DEL CONOCIMIENTO

Comprendimiento del  
panorama

**5**

## WEB SCRAPING

Cómo extraer los datos  
*ofuscados*

# BASE DE DATOS... ¿OFUSCADAS?

¿Qué es una Base de Datos?  
Cuál es su propósito



# BASE DE DATOS... ¿OFUSCADAS?

---

---

¿Qué es una Base de Datos?

# BASE DE DATOS... ¿OFUSCADAS?

---

---

¿Qué es una Base de Datos?

- Repositorio estructurado de información

Cuál es su propósito

# BASE DE DATOS... ¿OFUSCADAS?

---

¿Qué es una Base de Datos?

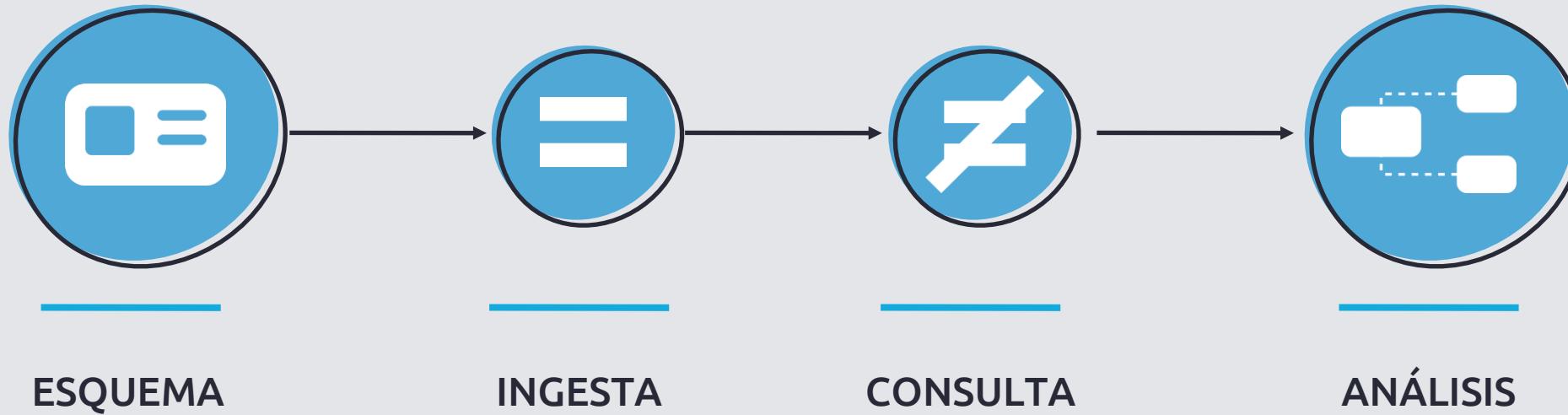
- Repositorio estructurado de información

Cuál es su propósito

- Almacenar información de manera estructurada
- Mutar información estructurada
- Consultar información estructurada

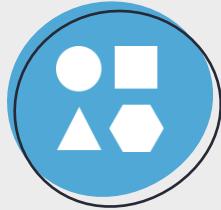
# ESTRUCTURA DE UNA BASE DE DATOS

---



# CONCEPTOS DE BASES DE DATOS

---



## ESQUEMA

Estructura de los datos definida con DDL (CREATE, ALTER, etc.)



## CONSULTA

Recabar información de los datos estructurados, definido con DML (SELECT)



## INGESTA

Acción sobre los datos definida con DML (INSERT, UPDATE, etc.)

## RECORDEMOS QUE

- DDL -> Data Definition Language (instrucciones para el esquema)
- DML -> Data Manipulation Language (instrucciones para las operaciones en un esquema)

# MUNDO DE DATOS

---

El siglo XXI es el siglo de la información

Las empresas compiten por tener datos, propios y externos

- Si sabes la estrategia de tus competidores la puedes rebatir

# DATOS OFUSCADOS



# DATOS OFUSCADOS

---

Por qué ofuscarían los datos las empresas

- E-commerce
- ¿.zip con toda la información comercial?

Mostrar información a usuarios, y entorpecérsela a competidores

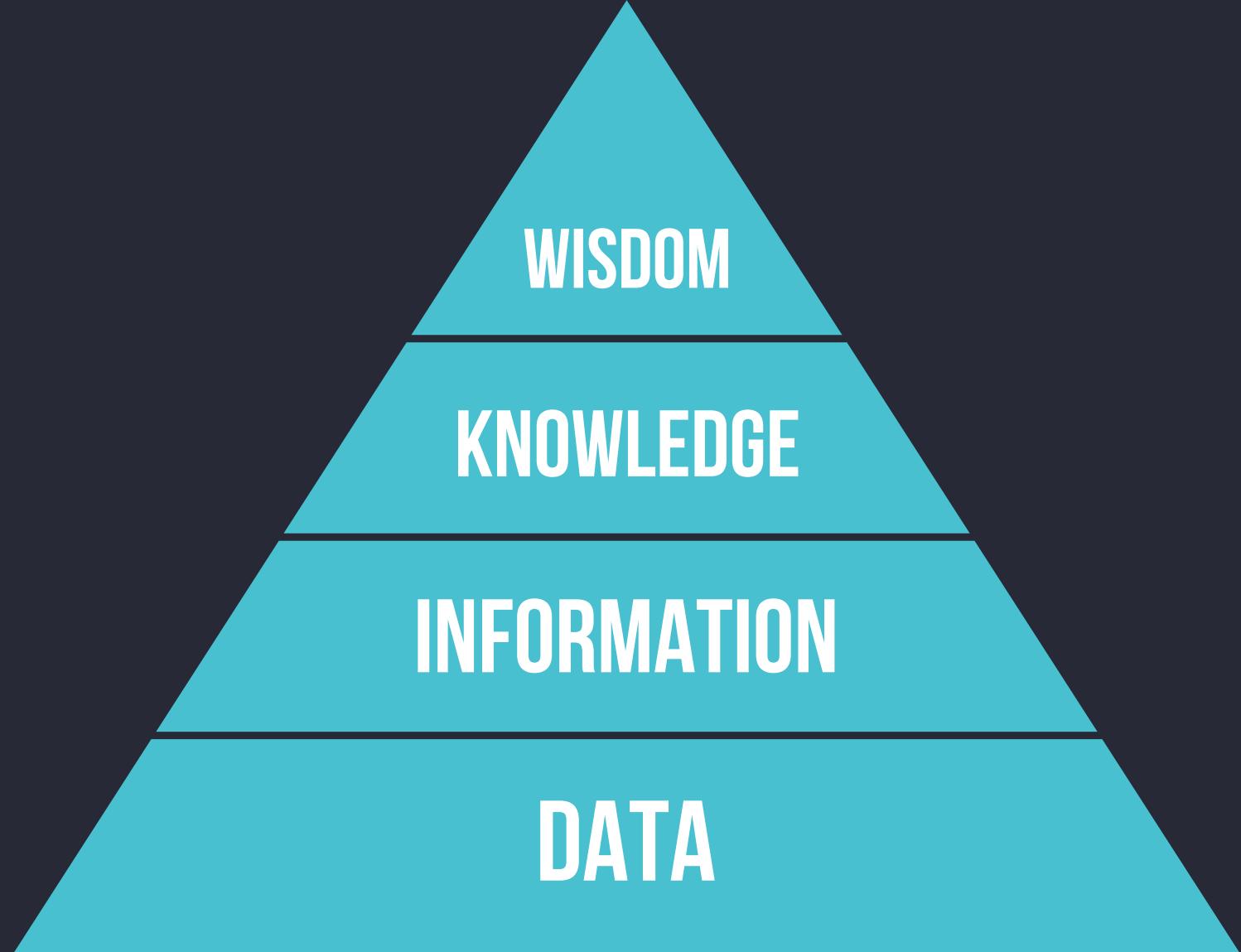
# PARA QUÉ NECESITAMOS DATOS

---

- Análisis de Datos
- Machine Learning
- Seguimiento de ofertas
- Comprender mejor un dominio

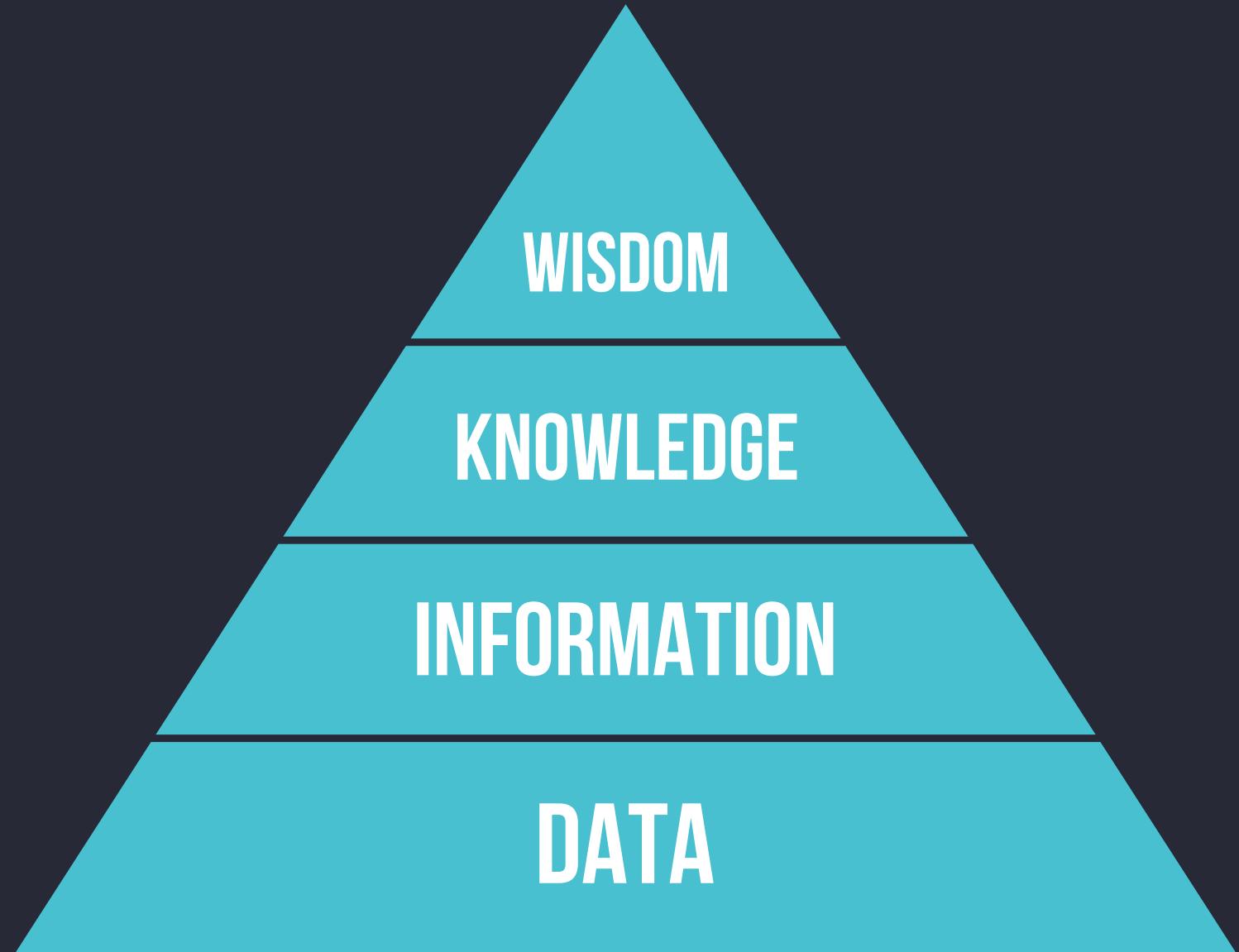


# PIRÁMIDE DEL CONOCIMIENTO



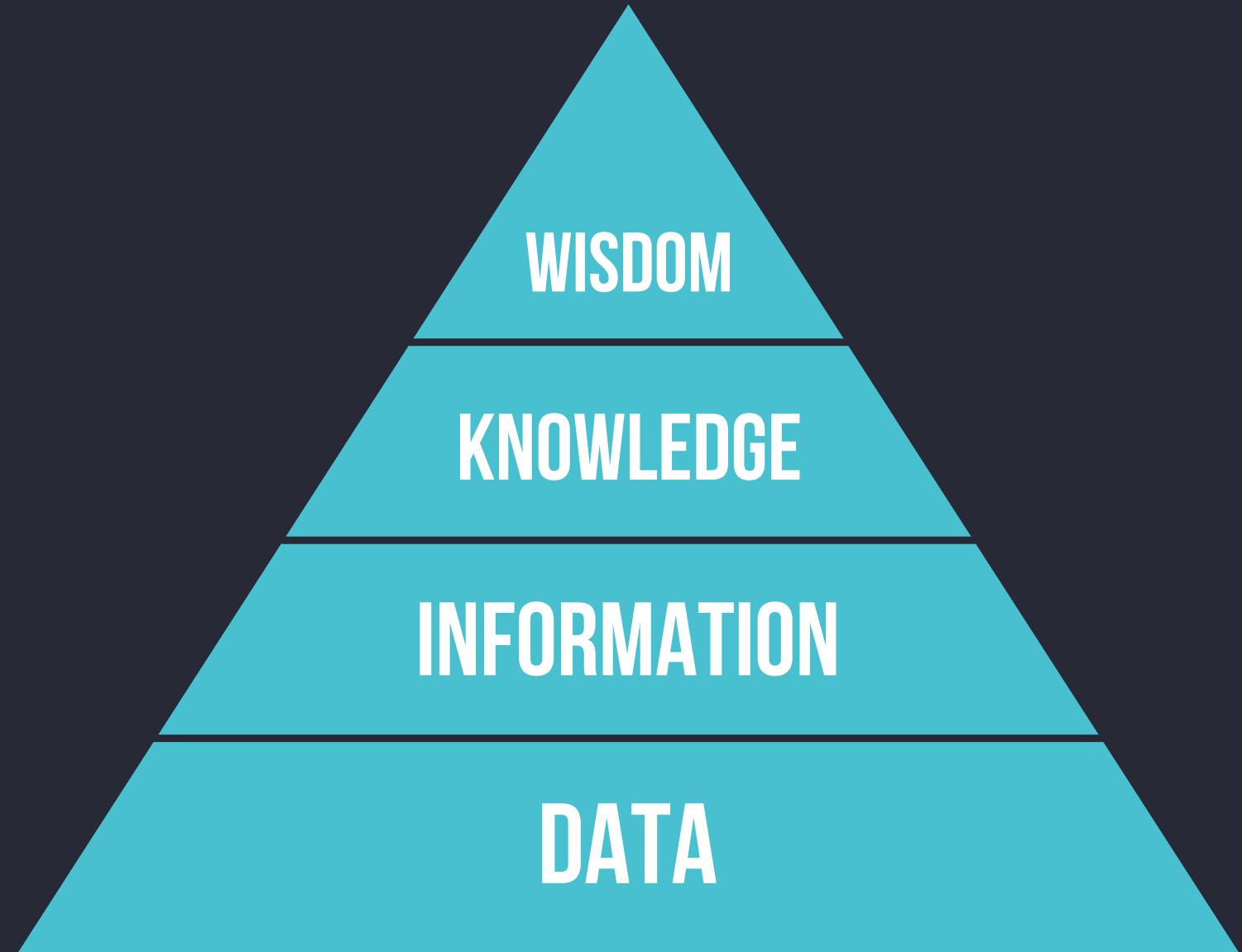


# PIRÁMIDE DEL CONOCIMIENTO





# PIRÁMIDE DEL CONOCIMIENTO



Información

Dato

- Valor puro



# PIRÁMIDE DEL CONOCIMIENTO

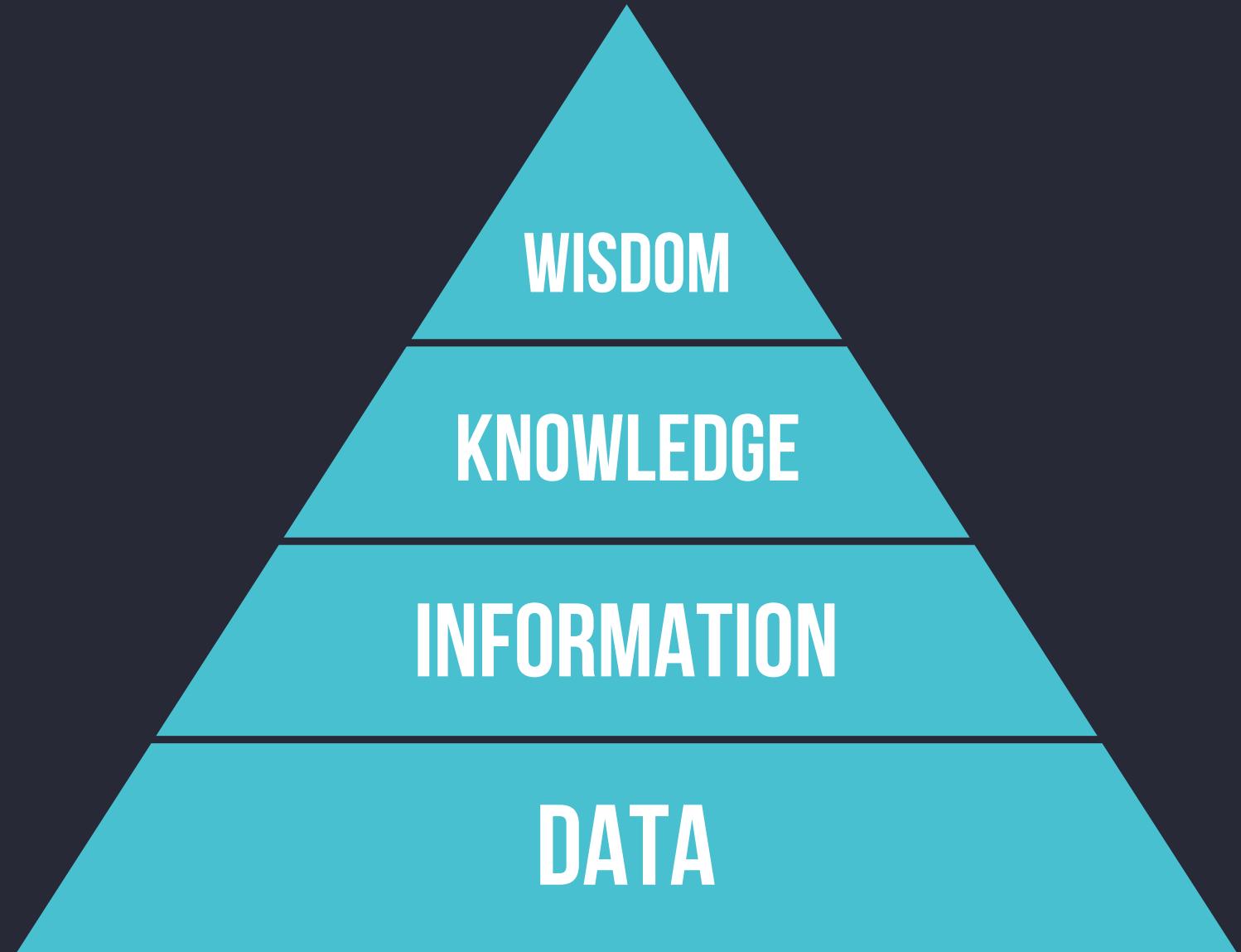
Conocimiento

Información

- Dato con contexto

Dato

- Valor puro





# PIRÁMIDE DEL CONOCIMIENTO

Sabiduría

Conocimiento

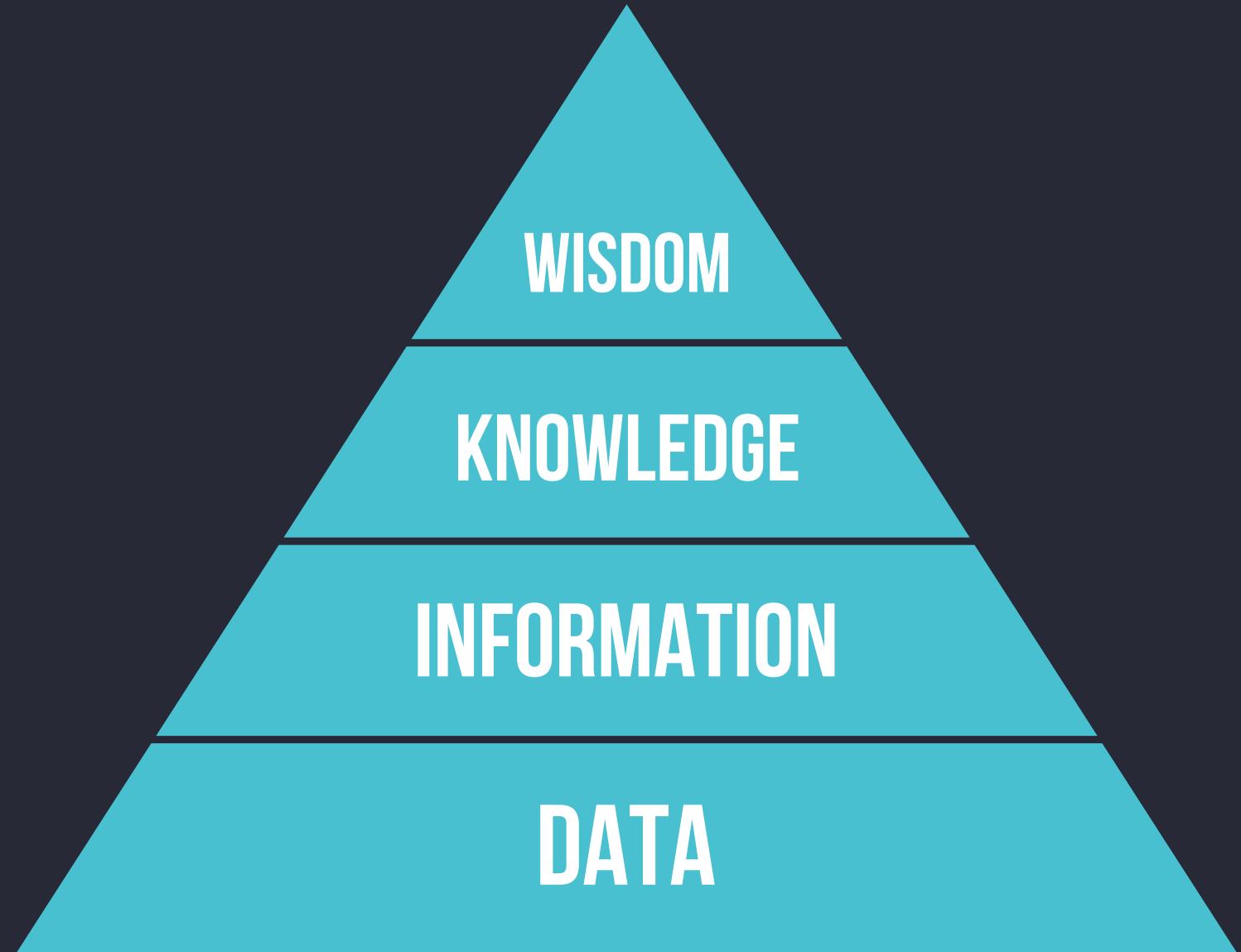
- Conjunto de información de un dominio
- Toma de decisiones aquí

Información

- Dato con contexto

Dato

- Valor puro





# PIRÁMIDE DEL CONOCIMIENTO

## Sabiduría

- Conjunto de conocimiento
- Predicción de comportamientos aquí

## Conocimiento

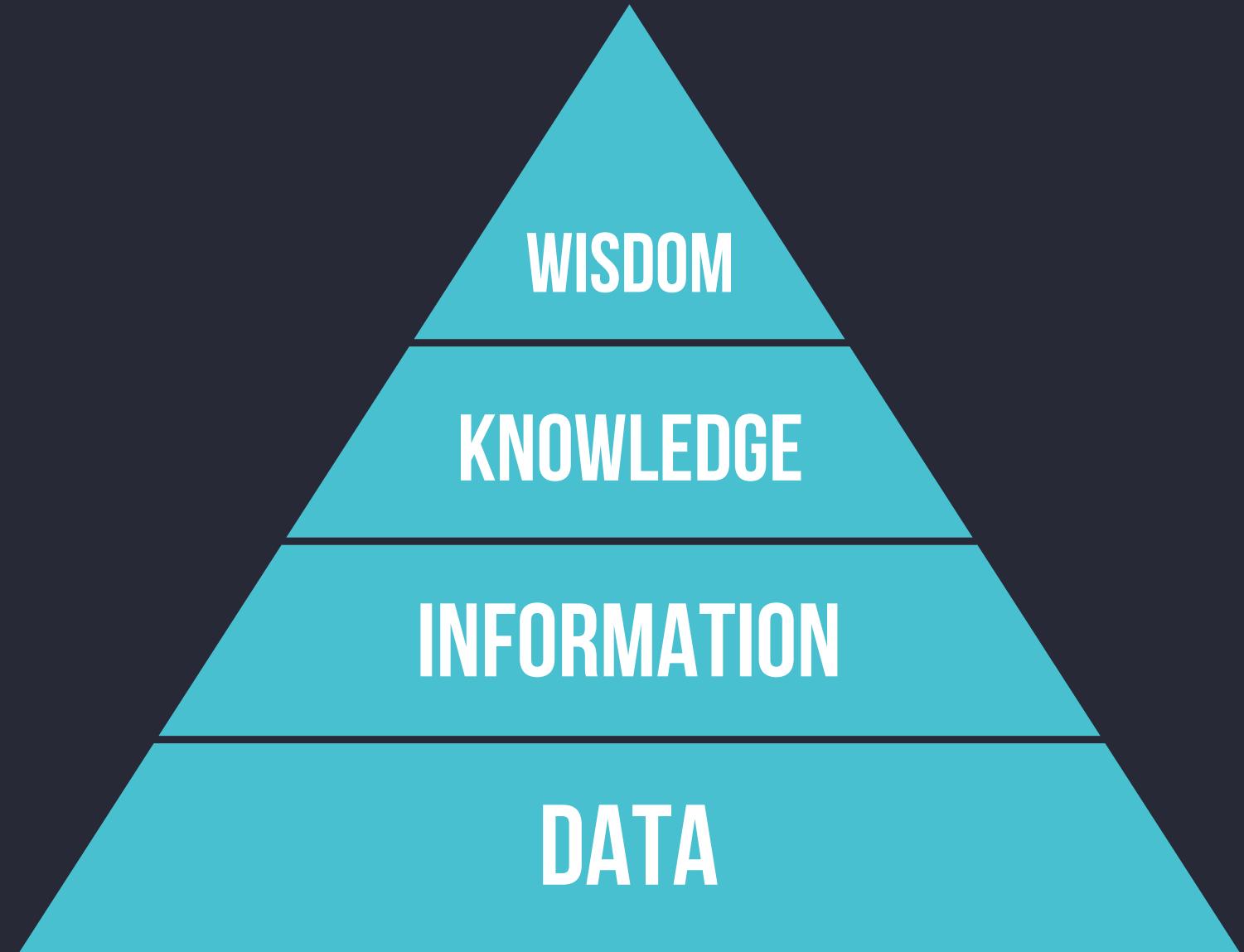
- Conjunto de información de un dominio
- Toma de decisiones aquí

## Información

- Dato con contexto

## Dato

- Valor puro





2

## WEB SCRAPING

---

# QUÉ ES WEB SCRAPING

---

**Extraer información de una página web de manera  
programática**

# UTILIDADES DEL WEB SCRAPING

---

Alertas personalizadas para sitios sin suscripciones

Avisos personalizados de descuentos en productos

Seguimiento y análisis de datos

Automatización de descargas con esquemas similares

- Moodle, Aules, Campus, etc.

# ESTADOS HTTP

---

1XX – Información

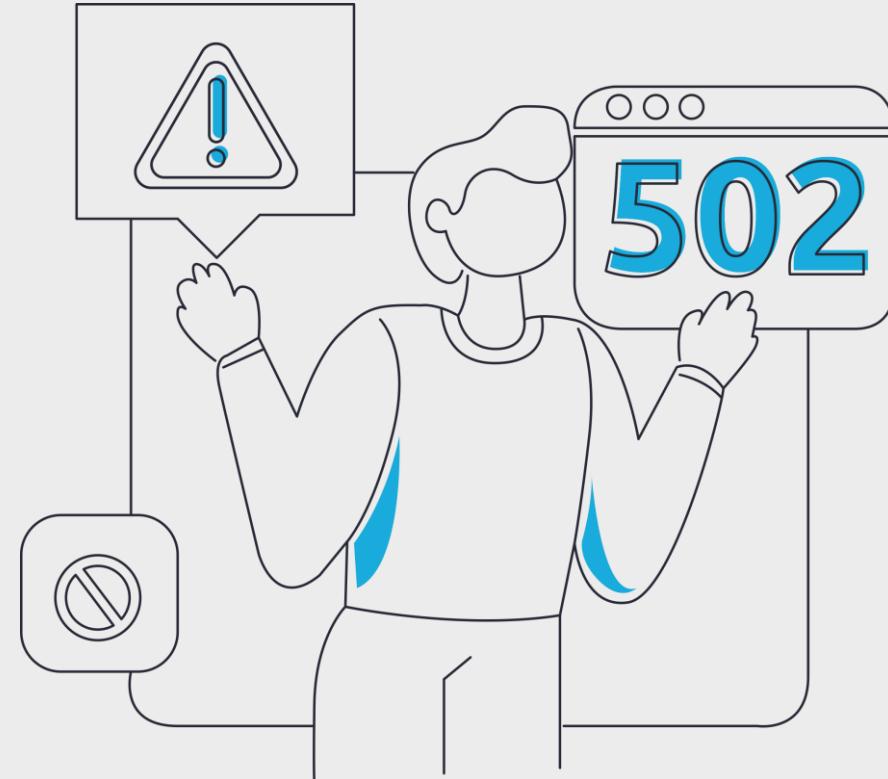
2XX – Éxito

3XX – Redirección

4XX – Fallo del cliente

- 418 Soy una tetera – Día de los inocentes

5XX – Fallo del servidor



Fuente: <https://umbraco.com/knowledge-base/http-status-codes/#:~:text=The%20100%20Continue%20status%20code,the%20request%20has%20already%20finished.>

# MÉTODOS HTTP

---

---

**GET**

POST

PUT

PATCH

DELETE

OPTIONS

... y más

# PIPELINE

Procedimiento habitual



# PIPELINE

---

---

## 1. Request

# PIPELINE

---

1. Request
  - Utilizando el User Agent correspondiente, solicitamos el contenido estático
2. Parser

# PIPELINE

---

1. Request
  - Utilizando el User Agent correspondiente, solicitamos el contenido estático
2. Parser
  - Parseamos el contenido utilizando XML, HTML, LXML, a elección
3. Extract

# PIPELINE

---

## 1. Request

- Utilizando el User Agent correspondiente, solicitamos el contenido estático

## 2. Parser

- Parseamos el contenido utilizando XML, HTML, LXML, a elección

## 3. Extract

- Realizamos queries para extraer la información relevante de la página

## 4. [Iterate]

# PIPELINE

---

1. Request
  - Utilizando el User Agent correspondiente, solicitamos el contenido estático
2. Parser
  - Parseamos el contenido utilizando XML, HTML, LXML, a elección
3. Extract
  - Realizamos queries para extraer la información relevante de la página
4. [Iterate]
  - Puede que tengamos que navegar la página (categorías, ofertas, etc.)
  - Puede que tengamos un listado de productos paginado

# EXTRACCIÓN

- Origen
- Métodos



# EXTRACCIÓN

---

- Sitemap
- Navegación
- Interacción

# EXTRACCIÓN POR SITEMAP

---

El sitemap es un XML usado para el SEO, puede haber múltiples sitemaps, por idiomas, imágenes, recursos, son configurables y comunicados a los navegadores (Google Search Console)

Lo que hace el sitemap es facilitarle la faena el robot de SEO, es decir, al servicio que se encarga de hacer web scraping

# EXTRACCIÓN POR NAVEGACIÓN

---

Navegando el header, extrayendo información del menú (categorías), buscando una página de secciones.

Navegando subsecciones a partir de X secciones

# EXTRACCIÓN POR INTERACCIÓN

---

Es parecida a la **Extracción por navegación**, pero con pasos extra, no siempre estará disponible todo lo que tienes que navegar, y, es más, dependerá de las acciones que programes que ciertos flujos de navegación estén habilitados o no.

# MÉTODOS DE LOCALIZACIÓN DE NODOS

---

XML

# MÉTODOS DE LOCALIZACIÓN DE NODOS

---

## XML

- Navegación por jerarquía de nodos, xPath simplificado

## xPath

# MÉTODOS DE LOCALIZACIÓN DE NODOS

---

## XML

- Navegación por jerarquía de nodos, xPath simplificado

## xPath

- Lenguaje de consultas de XML, más costoso de leer y de mantener, pero más versátil

## querySelector

# MÉTODOS DE LOCALIZACIÓN DE NODOS

---

## XML

- Navegación por jerarquía de nodos, xPath simplificado

## xPath

- Lenguaje de consultas de XML, más costoso de leer y de mantener, pero más versátil

## querySelector

- “Lenguaje” de localización de HTML mediante reglas de CSS, xPath algo menos versátil pero más legible

# Y DESPUÉS... ¿QUÉ?

---

- Analizar datos
- Transformar datos
- Almacenar en una BDD (MySQL, MariaDB, PostgreSQL, etc.)
- Almacenamiento físico estructurado (csv, xml, xlsx, etc.)

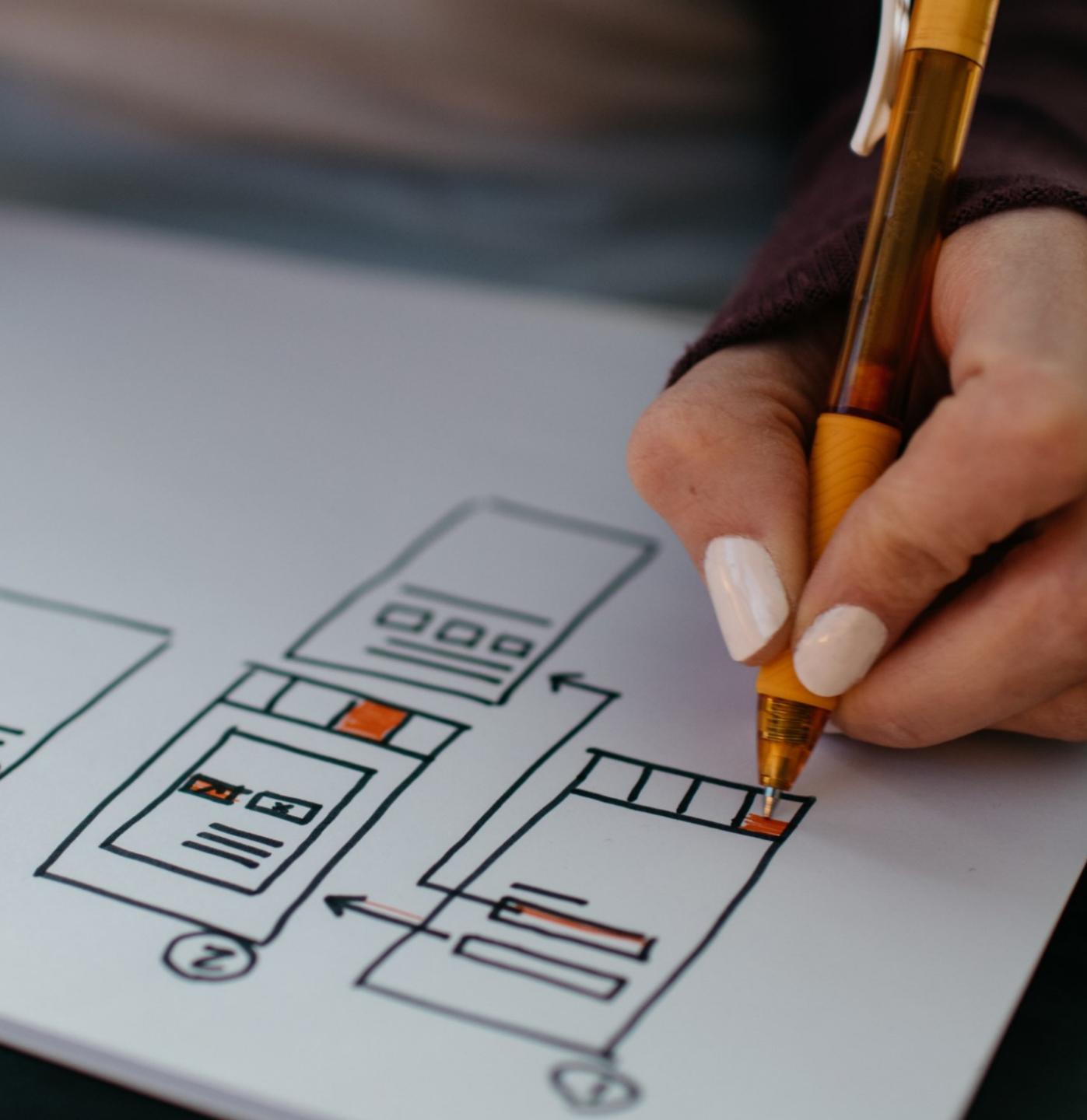
# DEMO

---

Página de productos sencilla

Tiempo aproximado: **10 minutos**

Target: **DRUNI**



3

## HEADLESS WEB SCRAPING

---

# HEADLESS WEB SCRAPING

---

Qué es un Headless Browser

# HEADLESS WEB SCRAPING

---

Qué es un Headless Browser

- Un navegador que puede funcionar programáticamente (sin usuario)

# HEADLESS WEB SCRAPING

---

Qué es un Headless Browser

- Un navegador que puede funcionar programáticamente (sin usuario)

Ejemplos

- Chromium (el engine)
- Selenium

# QUÉ NOS OFRECE

---

---

Conexión ininterrumpida

# QUÉ NOS OFRECE

---

## Conexión ininterrumpida

- No es un HTTP GET, es una conexión que no se cierra hasta que queramos

## Simular ser un usuario

# QUÉ NOS OFRECE

---

Conexión ininterrumpida

- No es un HTTP GET, es una conexión que no se cierra hasta que queramos

Simular ser un usuario

- Interacciones, user agent, tracking y sesiones automáticas

Interactuar con la página programáticamente

# QUÉ NOS OFRECE

---

Conexión ininterrumpida

- No es un HTTP GET, es una conexión que no se cierra hasta que queramos

Simular ser un usuario

- Interacciones, user agent, tracking y sesiones automáticas

Interactuar con la página programáticamente

- Clics, delays, esperar a que cargue el DOM y scripts

# ¿POR QUÉ NECESITAMOS HEADLESS?

---

Frameworks

SPAs

Interacciones con la página

# FRAMEWORKS Y LIBRERÍAS

---

Scope de classNames

classNames autogenerados/ofuscados en build-time

Componentes con información en memoria esperando interacciones

NextJS y Remix, serialización del servidor y SSR

# SPA

---

Carga estática de un HTML simple sin información

- Pantallazo blanco
- CSR -> el contenido se genera tras la carga

La página carga con el DOM listo

Requiere de interacciones y enrutaciones para acceder al contenido de verdad

# INTERACCIONES CON LA PÁGINA

---

Páginas que no usar URL as State Manager

- Paginaciones
- Filtros

Cálculos al vuelo e información consultada con servidor

- Página oficial de la lotería
- Calculadoras de nóminas, hipotecas, etc.

Paginaciones dinámicas, en base a respuestas del servidor

Paginaciones por cursor (no hay limit ni offset)

Infinite scrolling para listados

# OPCIONES DE HEADLES SCRAPING

---

- Selenium -> ampliamente conocido
- Puppeteer
- Cypress
- Playwright

## Alternativamente...

Consola de la página y JavaScript puro

- Útil para ejecuciones manuales y poco alcance

# HERRAMIENTAS

Google Colab

# HERRAMIENTAS

## Google Colab

- Jupyter Notebook en la nube
- Acceso gratuito para bajo rendimiento

## Jupyter Notebook

# HERRAMIENTAS

## Google Colab

- Jupyter Notebook en la nube
- Acceso gratuito para bajo rendimiento

## Jupyter Notebook

- Documentación y código en un mismo fichero
- Ejecución de bloques de código con contexto compartido

## Ventajas de usar Google Colab

# HERRAMIENTAS

## Google Colab

- Jupyter Notebook en la nube
- Acceso gratuito para bajo rendimiento

## Jupyter Notebook

- Documentación y código en un mismo fichero
- Ejecución de bloques de código con contexto compartido

## Ventajas de usar Google Colab

- IP y VPN externas, no nos limitaremos por proxies
- Reiniciar entorno y generar nueva IP (por si nos bloquean)
- Internet y ancho de banda más alto
- Almacenamiento directo en la nube

# DEMO: HEADLESS WEB SCRAPING

---

Páginas más “ofuscadas”, (in)voluntariamente, y cómo un headless browser nos ayuda

Tiempo aproximado: **10 minutos**

Target: **ZARA**

# SI UN USUARIO LO VE, LO PUEDES SCRAPEAR

Toda información accesible a un usuario, puede ser extraída programáticamente



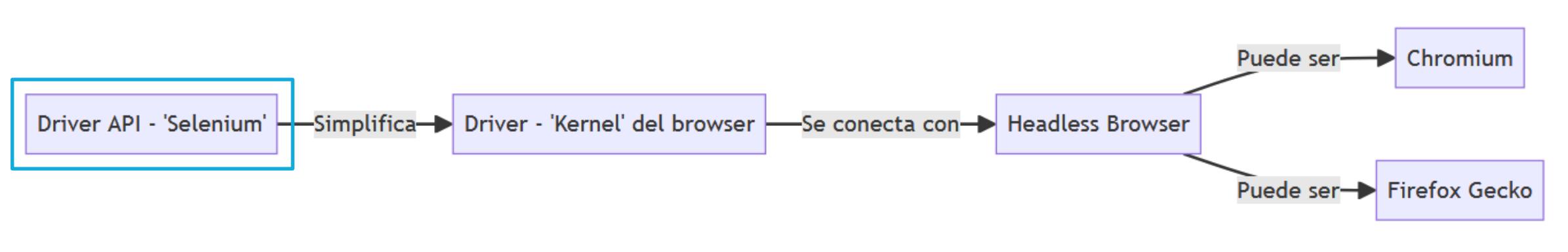


4

## SELENIUM

# ARQUITECTURA DE SELENIUM

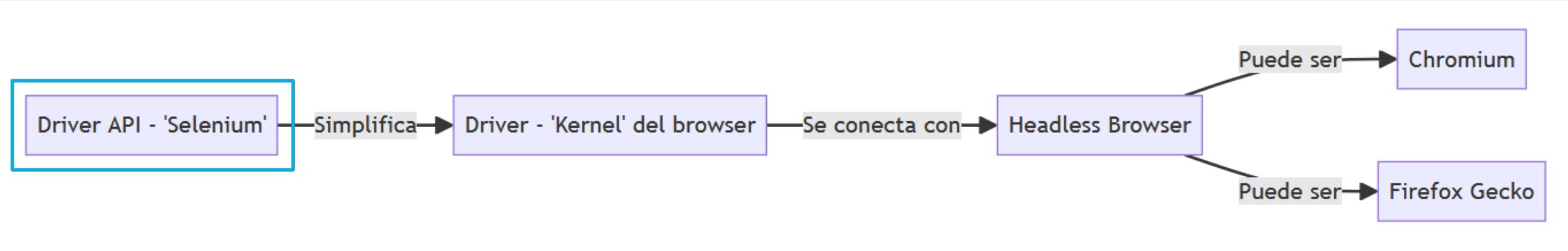
---



La API es dependiente del lenguaje que se esté utilizando

# ARQUITECTURA DE SELENIUM

---

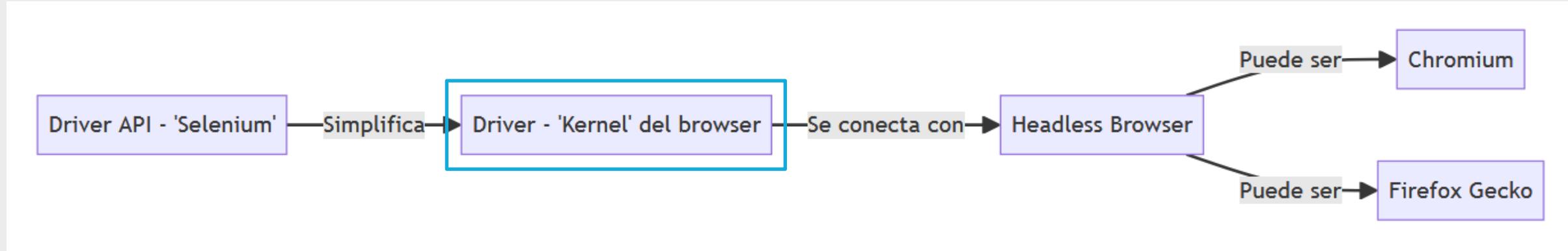


La API es dependiente del lenguaje que se esté utilizando

1. La librería podrá cambiar sintaxis aún manteniendo conceptos
2. El funcionamiento será muy parecido, pero con las peculiaridades de cada lenguaje (Python GIL, Java JVM)

# ARQUITECTURA DE SELENIUM

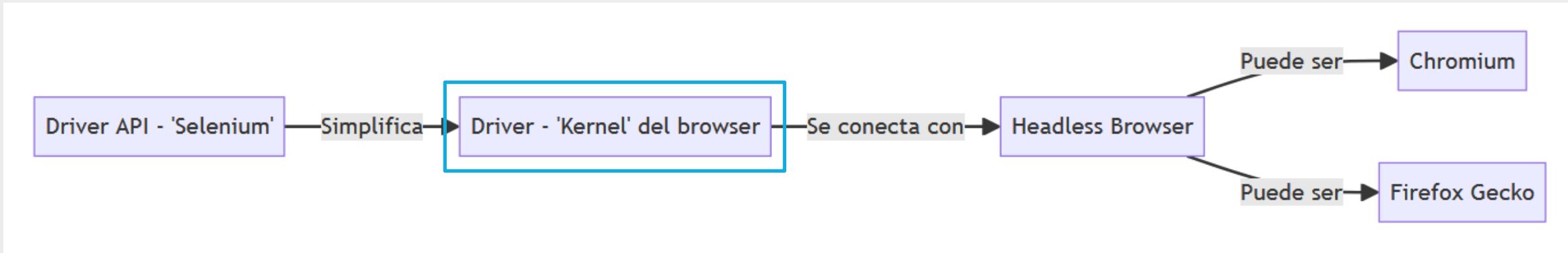
---



El driver es la conexión específica de la librería con el headless browser, es el elemento común

# ARQUITECTURA DE SELENIUM

---

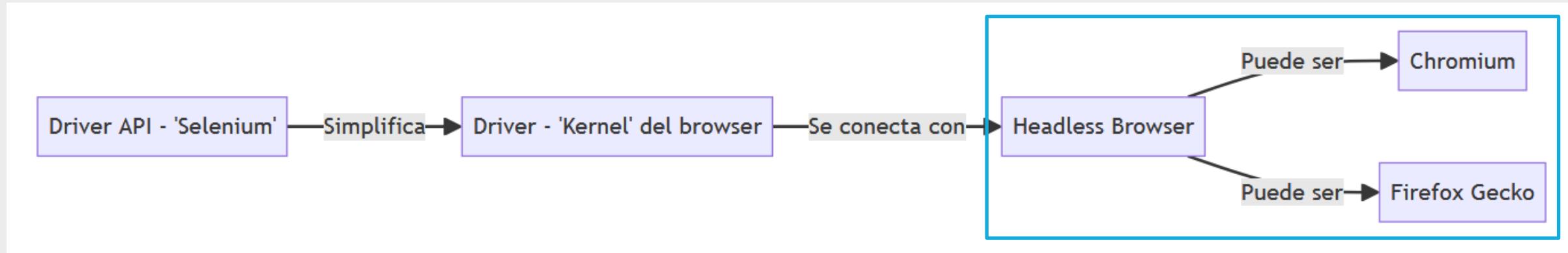


El driver es la conexión específica de la librería con el headless browser, es el elemento común

1. Java usa un JVM para proporcionar un lenguaje unificado sin importar el Sistema Operativo
2. Selenium tiene un driver que se conectará con cada navegador (o "será" el navegador)

# ARQUITECTURA DE SELENIUM

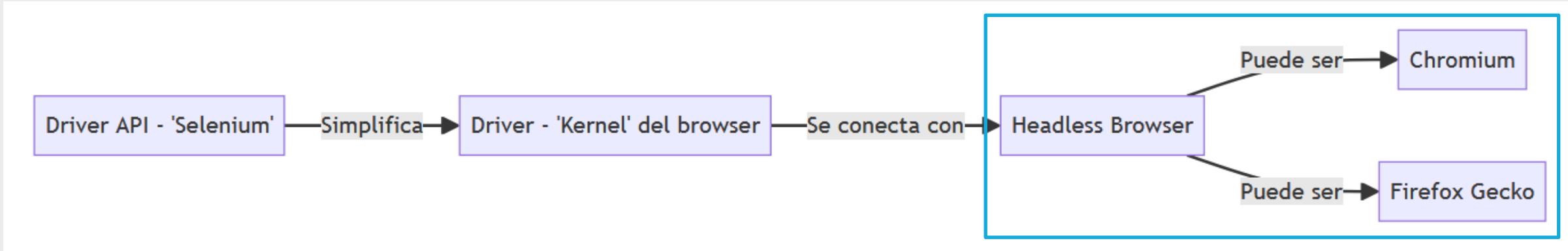
---



Headless Browser, el *flavor* del navegador

# ARQUITECTURA DE SELENIUM

---



Headless Browser, el *flavor* del navegador tiene diferencias y soportes, user-agents diferentes, motores diferentes

1. Unificando conceptos (driver) y con una librería que en varios lenguajes, es lógico no atarse a un navegador, ahí entran los engines que queramos usar
2. Google Chrome usa el motor de Chromium (y la mayoría de los navegadores también), Firefox tiene su propio motor (Gecko), y Safari también (WebKit)

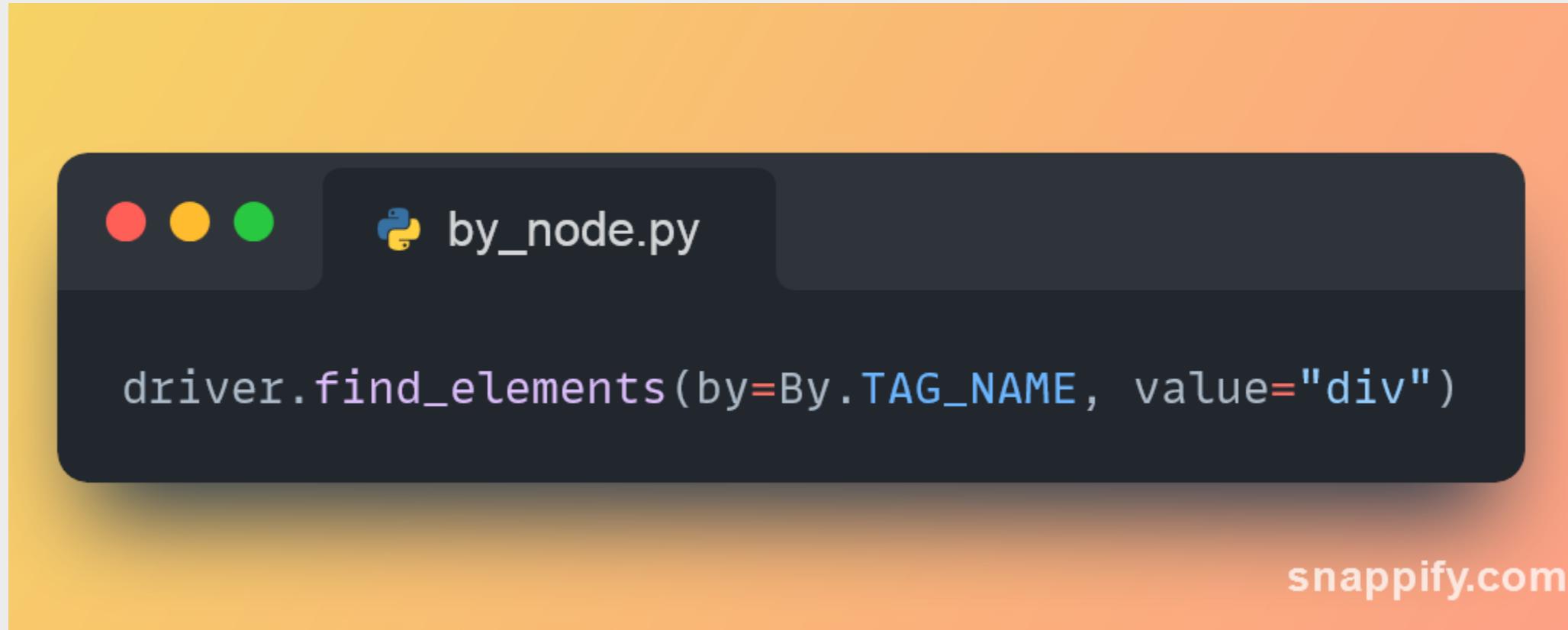
# API DE SELENIUM

---

- Buscar
- Interactuar
- Capturar

# BUSCAR ELEMENTOS POR NODO

---



# BUSCAR ELEMENTOS POR CLASE

---



by\_class.py

```
driver.find_elements(by=By.CLASS_NAME, value="aaaaaa")
```

[snappify.com](https://snappify.com)

# BUSCAR ELEMENTOS POR ID

---



by\_id.py

```
driver.find_elements(by=By.ID, value="aaaaa")
```

snappify.com

# BUSCAR ELEMENTOS POR QUERYSELECTOR

---



by\_querySelector.py

```
driver.find_elements(by=By.CSS_SELECTOR, value="div > img")
```

snappify.com

# RELENAR UN FORMULARIO

---



form\_interaction.py

```
element = driver.find_element(by=By.CSS_SELECTOR, value="textarea")
element.send_keys("Envía cada una de las teclas como interacción de usuario")
```

snappify.com

# INTERACCIÓN CON CLICK

---



click\_interaction.py

```
element = driver.find_element(by=By.CSS_SELECTOR, value="button")
element.click()
```

snappify.com

# INTERACCIÓN CON SCROLL

---



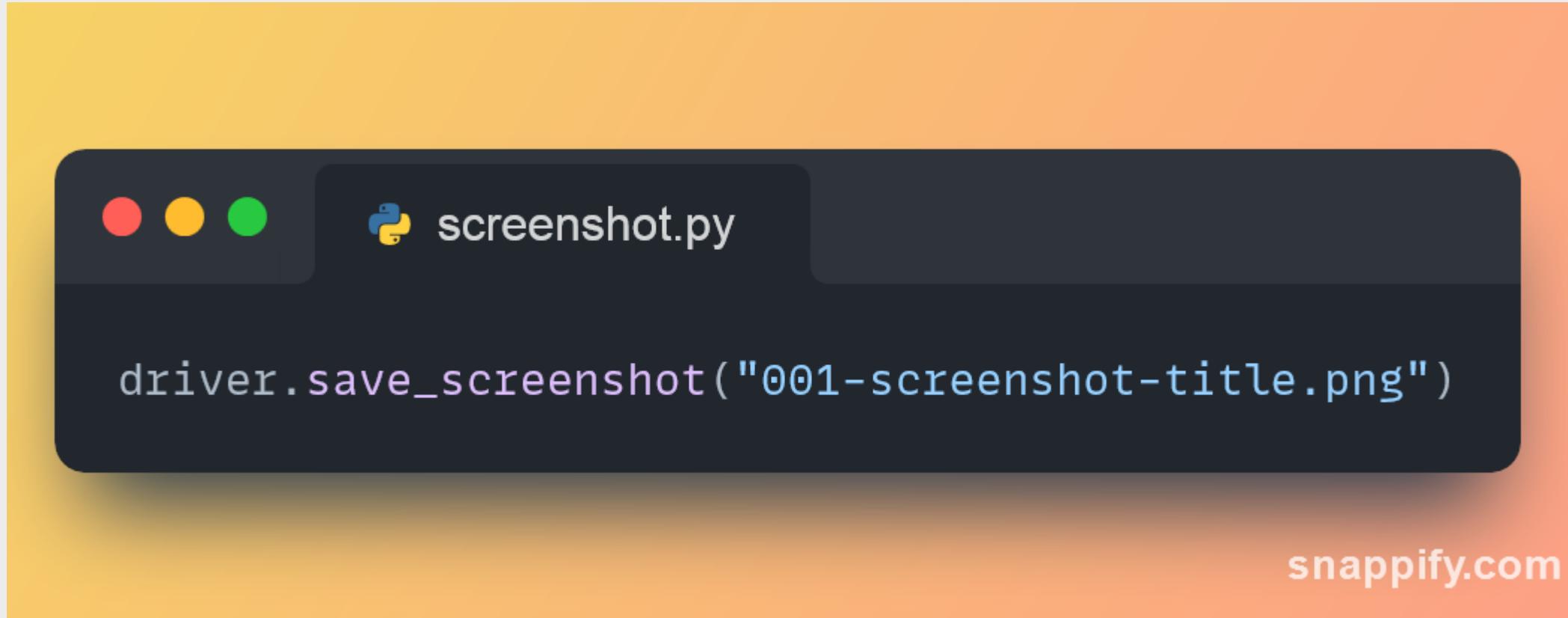
scroll\_interaction.py

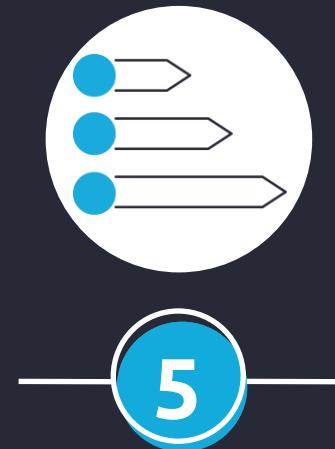
```
from selenium.webdriver import ActionChains  
  
amount = ActionChains(driver).scroll_by_amount(0, amount).perform()
```

[snappify.com](https://snappify.com)

# CAPTURAS DE PANTALLA

---





## DEFENSA ANTE WEB SCRAPING

# ¿CÓMO OFUSCAMOS NUESTROS DATOS?

---

---

“Defensa” ante web scraping

# QUÉ OCURRE CUANDO HACEMOS WEB SCRAPING PARA EL OTRO LADO

---

## Extracción de datos

- Clientes potenciales -> bien
- Competidores -> no tan bien

## DDoS involuntario

## Extracción de información confidencial (posibles exploits)

# TÉCNICAS DE “OFUSCACIÓN”

---

---

Throttling

# TÉCNICAS DE “OFUSCACIÓN”

---

## Throttling

- Peticiones con un mismo patrón de por medio (cada X segundos) son un robot, así que fuera DDoS protection (Cloudflare por ejemplo, delays)

# TÉCNICAS DE “OFUSCACIÓN”

---

## Throttling

- Peticiones con un mismo patrón de por medio (cada X segundos) son un robot, así que fuera DDoS protection (Cloudflare por ejemplo, delays)
- Delay en la información y registro de sesión

## Captcha (¿eres un robot?)

# TÉCNICAS DE “OFUSCACIÓN”

---

## Throttling

- Peticiones con un mismo patrón de por medio (cada X segundos) son un robot, así que fuera DDoS protection (Cloudflare por ejemplo, delays)
- Delay en la información y registro de sesión

## Captcha (¿eres un robot?)

- Requiere interacción avanzada (Computer visión + Interacción programática)

## Ofuscamiento de classNames (nuestra guía cambiante)

# TÉCNICAS DE “OFUSCACIÓN”

---

## Throttling

- Peticiones con un mismo patrón de por medio (cada X segundos) son un robot, así que fuera DDoS protection (Cloudflare por ejemplo, delays)
- Delay en la información y registro de sesión

## Captcha (¿eres un robot?)

- Requiere interacción avanzada (Computer visión + Interacción programática)

## Ofuscamiento de classNames (nuestra guía cambiante)

- Si extraemos información por classNames, autogenerarlos con hashes nos dificultaría la faena Agente web o User-Agent

# TÉCNICAS DE “OFUSCACIÓN”

---

## Throttling

- Peticiones con un mismo patrón de por medio (cada X segundos) son un robot, así que fuera DDoS protection (Cloudflare por ejemplo, delays)
- Delay en la información y registro de sesión

## Captcha (¿eres un robot?)

- Requiere interacción avanzada (Computer visión + Interacción programática)

## Ofuscamiento de classNames (nuestra guía cambiante)

- Si extraemos información por classNames, autogenerarlos con hashes nos dificultaría la faena
- Agente web o User-Agent
- Cabecera de peticiones, información de los navegadores

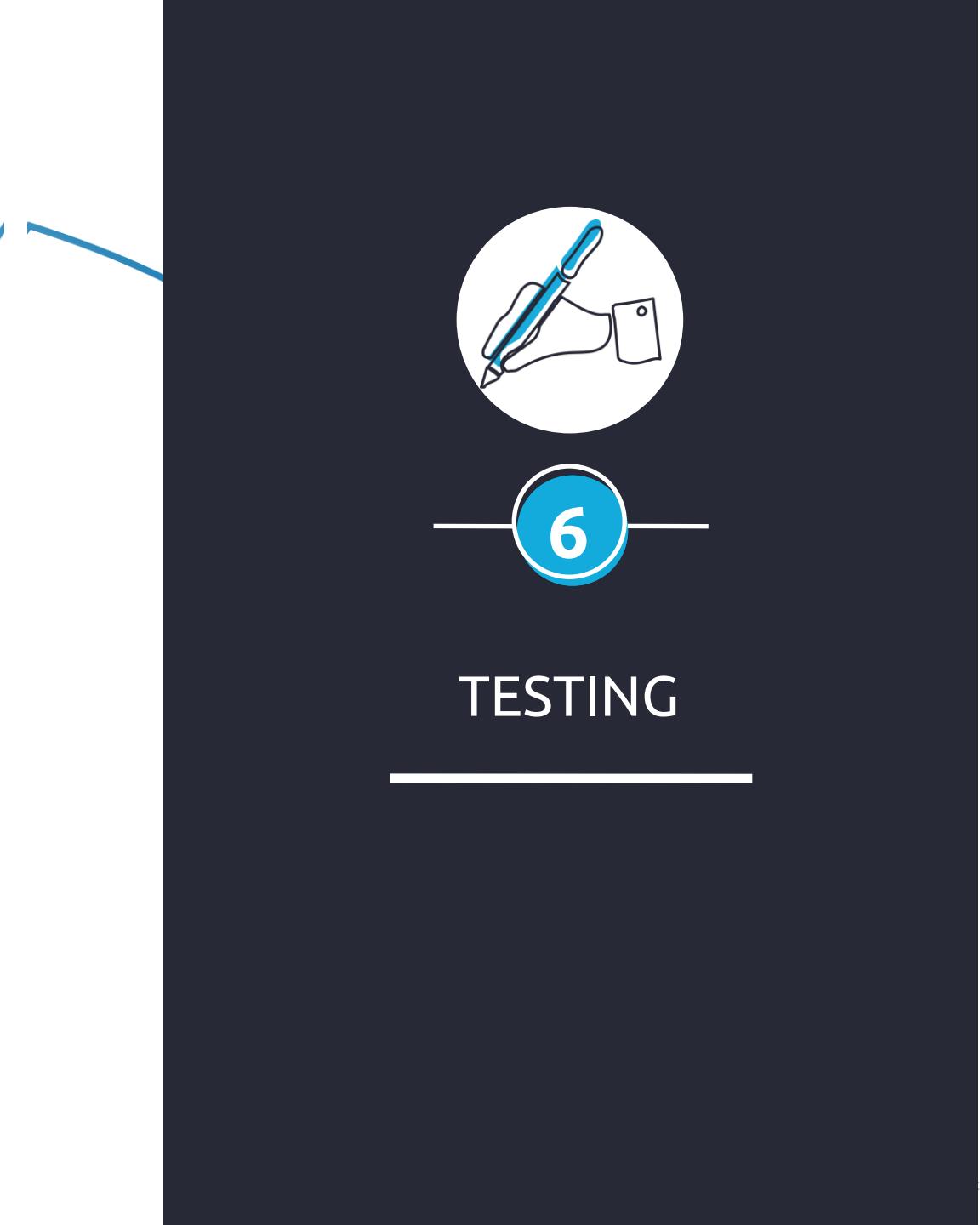
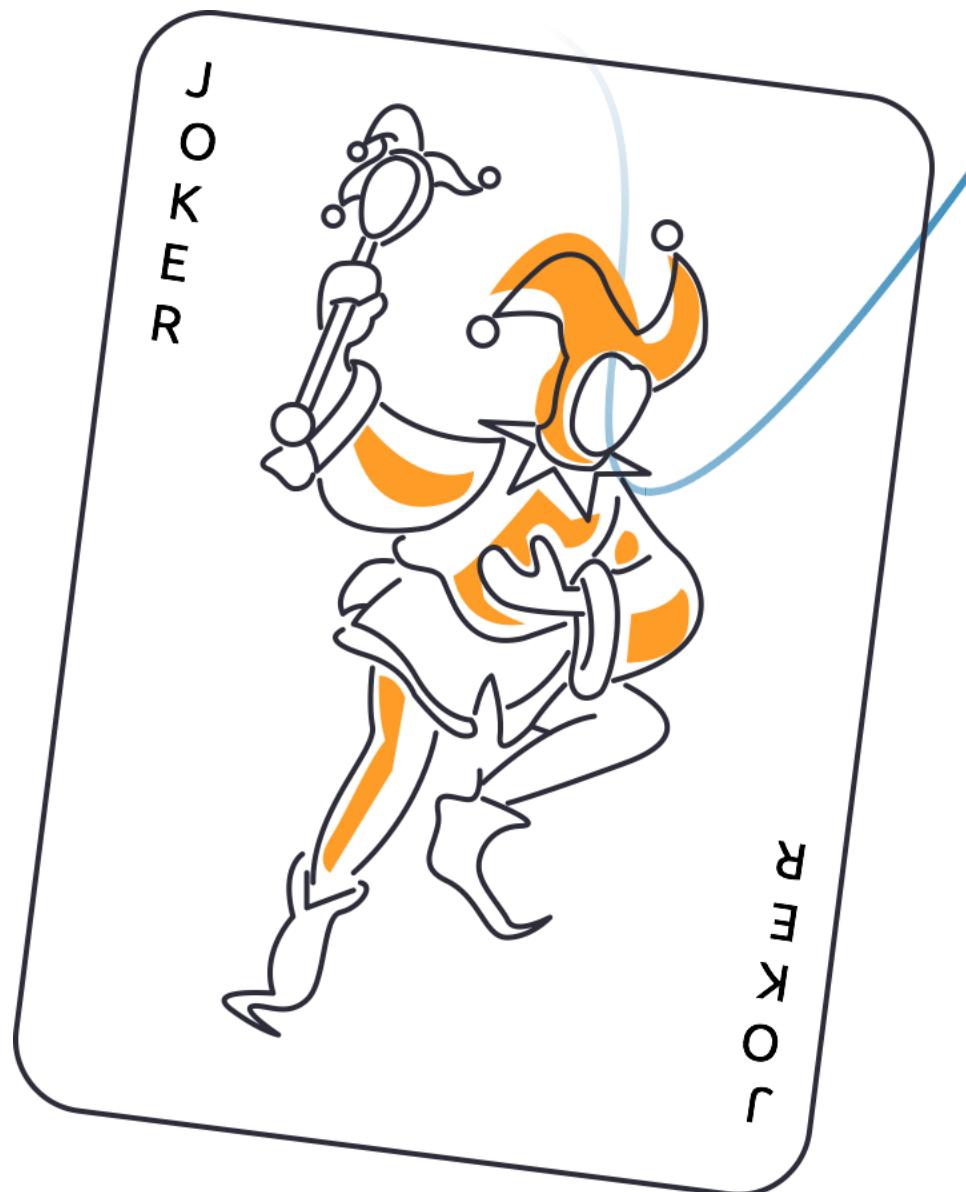
# DEMO: SIN HEADLESS

---

Páginas más “ofuscadas”, (in)voluntariamente, y los problemas que atañen

Tiempo aproximado: **5 minutos**

Target: **ZARA**



# TESTING

---

¿Qué es el testing? ¿Y qué propósito cumple?

# TESTING

---

¿Qué es el testing? ¿Y qué propósito cumple?

- La validación funcional de nuestro código
- Cumple el propósito de comprobar las funcionalidades una única vez

¿Es viable?

# TESTING

---

¿Qué es el testing? ¿Y qué propósito cumple?

- La validación funcional de nuestro código
- Cumple el propósito de comprobar las funcionalidades una única vez

¿Es viable?

- Todo código puede ser testeado (y automatizado)
- Qué partes deberían testearse y mantenerse es la clave

# ¿QUÉ TIPOS DE TESTS CONOCÉIS?

---

¿Quién se ánima a enumerar tipos de tests?

# TIPOS DE TESTS Y HERRAMIENTAS

---

Estáticos (Tipados, Clases Abstractas, Interfaces, Structs)

- IntelliSense

Unitarios

- JUnit, Mockito, Vitest, Enzyme

Integración

- JUnit, Testing Library, Vitest, Enzyme

End-to-end (e2e)

- Selenium, Cypress, PlayWright, Puppeteer, Postman

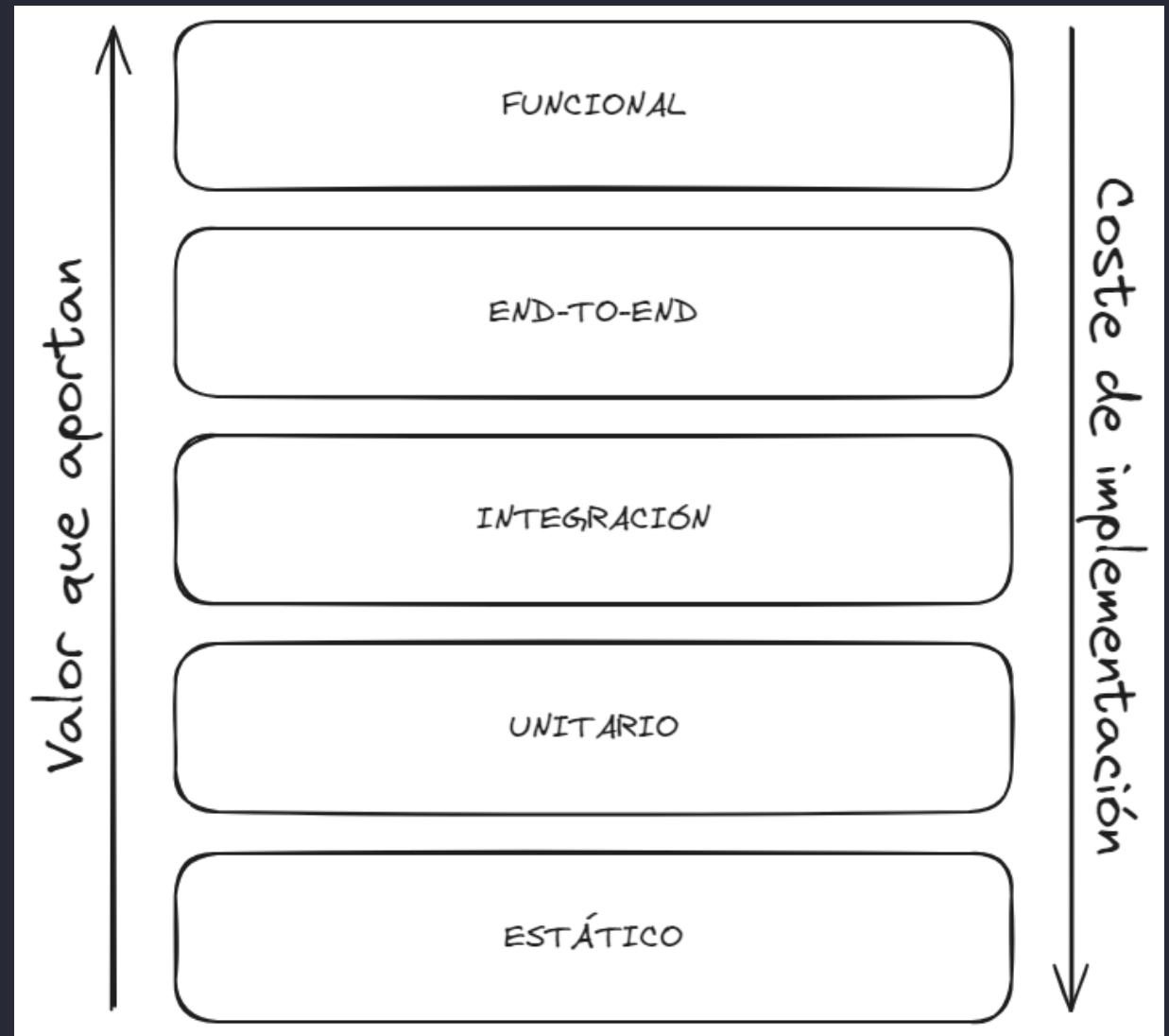
Smoke (Comprobaciones de sistema)

- ping, Kubernetes + Istio

...y unos cuantos más,



# JERARQUÍA DE TESTING



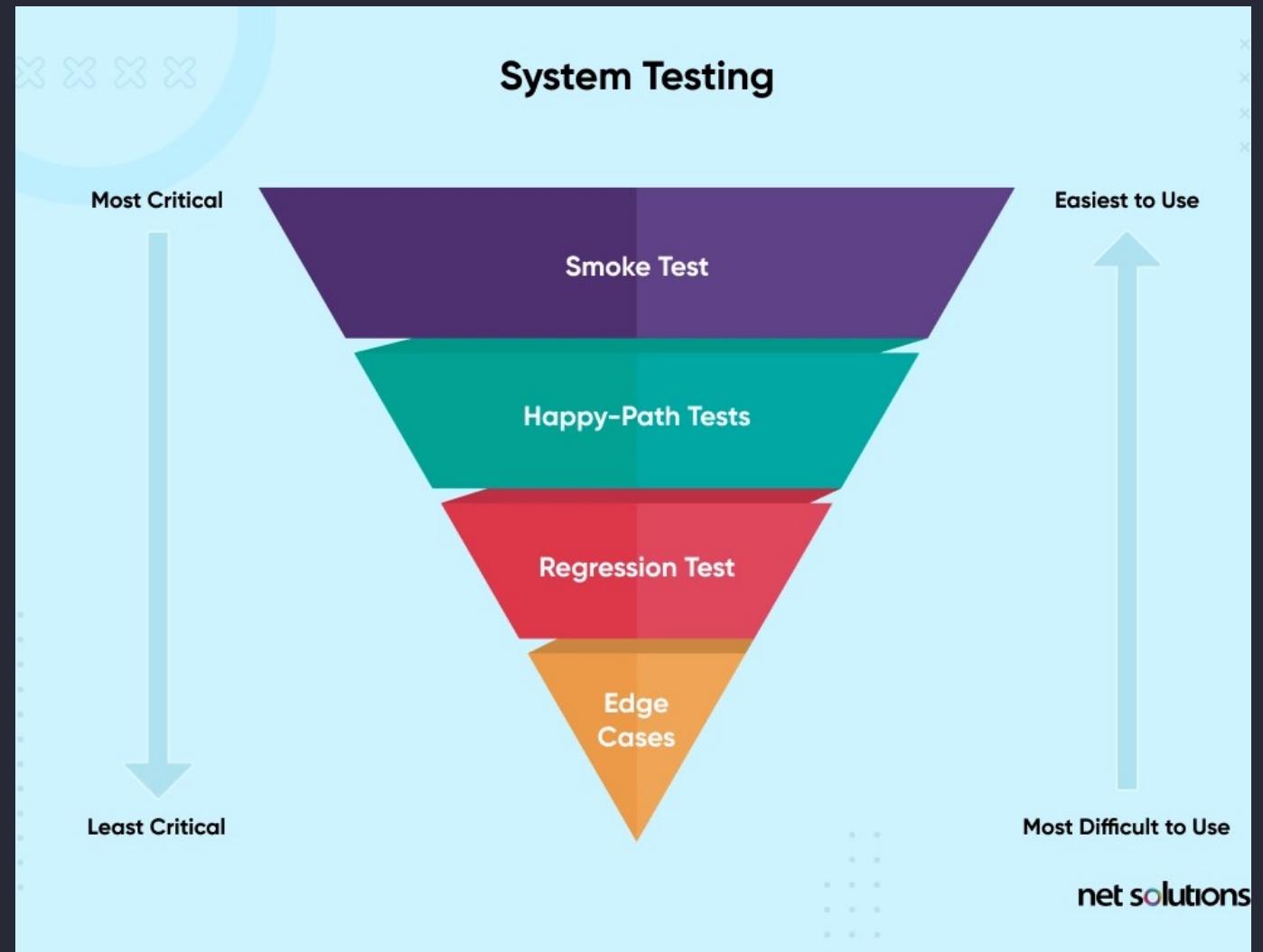


# END-TO-END

Cubrir una prueba de extremo a extremo

Son las más fiables y más costosas de mantener

Qué testearíamos y qué no





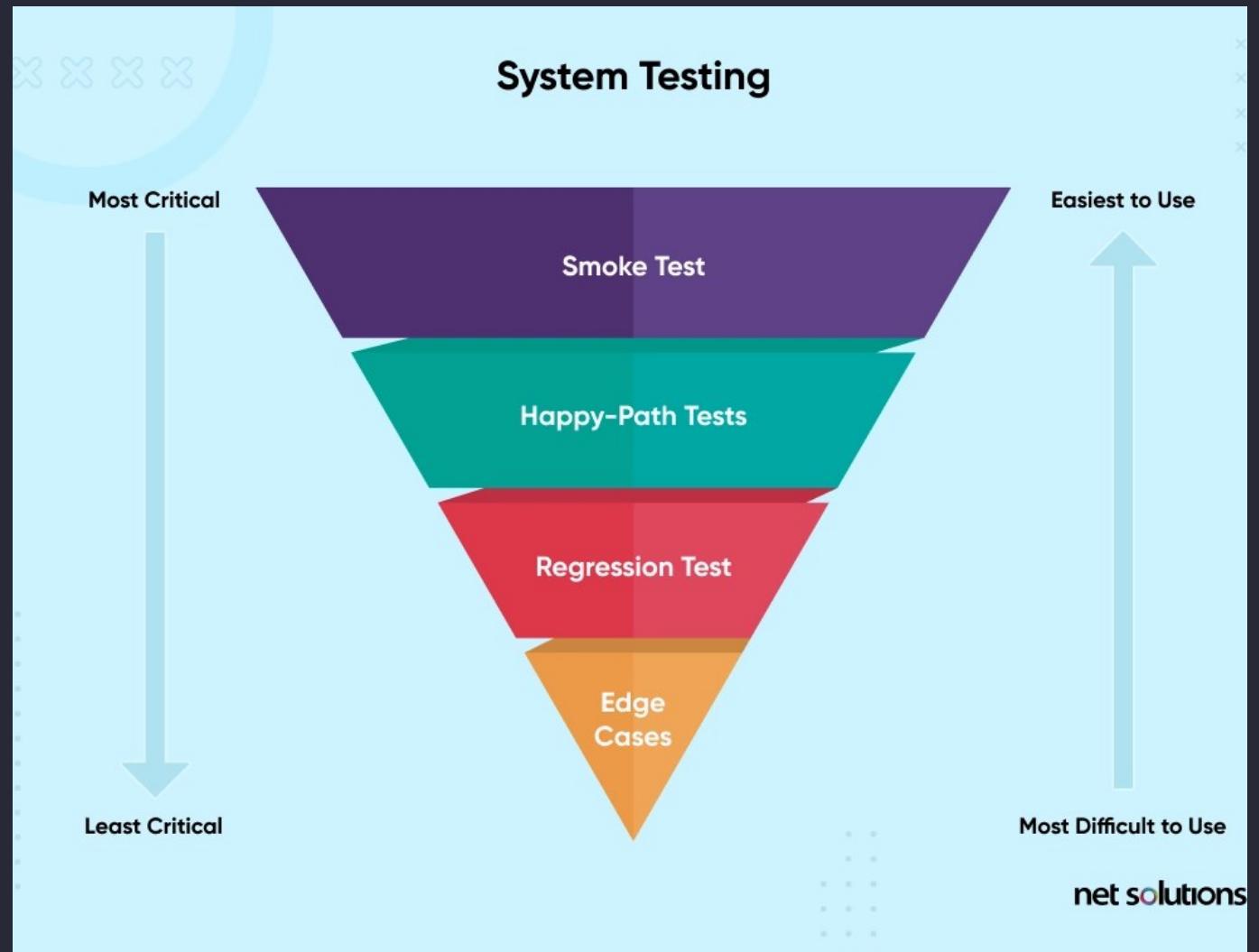
# END-TO-END

Cubrir una prueba de extremo a extremo

Son las más fiables y más costosas de mantener

Qué testearíamos y qué no

- El sistema funciona (HTTP GET)
- Página de producto fiable
- Recuperar los enlaces de categorías
- Página inexistente



# ¿SON NECESARIOS?

---

---

Los malos tests

# ¿SON NECESARIOS?

---

## Los malos tests

- Siempre será mejor no tener tests, que tests incompletos, o flaky

## Mantenimiento de los tests

# ¿SON NECESARIOS?

---

## Los malos tests

- Siempre será mejor no tener tests, que tests incompletos, o flaky

## Mantenimiento de los tests

- Los más valiosos serían e2e, si el proyecto y/o equipo es pequeño, tal vez no compense

## Empresas dedicadas

# ¿SON NECESARIOS?

---

## Los malos tests

- Siempre será mejor no tener tests, que tests incompletos, o flaky

## Mantenimiento de los tests

- Los más valiosos serían e2e, si el proyecto y/o equipo es pequeño, tal vez no compense

## Empresas dedicadas

- Si es la fuente principal de financiación, son necesarios, e incluso obligatorios
- Si una de las páginas de las que haces seguimiento cambia
  - Mejor enterarte tú por un test que rompe
  - Que al mes y no tener información

# REFERENCIA DE TESTING PARA WEB SCRAPING

---

<https://webscraper.io/test-sites>

Para los tests con web scraping, se recomienda usar las librerías que se utilizarían para un e2e, y para los tests unitarios lo mismo, en caso de Java, Junit, en el caso de Python, usar el *built-in assert*

7

## RECAPITULANDO

---

# RECAPITULANDO...

---

---

Qué hemos visto

# RECAPITULANDO...

---

---

## 1. Conceptos de una Base de Datos

# RECAPITULANDO...

---

---

1. Conceptos de una Base de Datos
2. Web Scraping y las Bases de Datos ofuscadas

# RECAPITULANDO...

---

1. Conceptos de una Base de Datos
2. Web Scraping y las Bases de Datos ofuscadas
3. Headless Web Scraping

# RECAPITULANDO...

---

---

1. Conceptos de una Base de Datos
2. Web Scraping y las Bases de Datos ofuscadas
3. Headless Web Scraping
4. Selenium

# RECAPITULANDO...

---

---

1. Conceptos de una Base de Datos
2. Web Scraping y las Bases de Datos ofuscadas
3. Headless Web Scraping
4. Selenium
5. Defensa ante Web scraping
  1. Riesgos del Web scraping

# RECAPITULANDO...

---

1. Conceptos de una Base de Datos
2. Web Scraping y las Bases de Datos ofuscadas
3. Headless Web Scraping
4. Selenium
5. Defensa ante Web scraping
  1. Riesgos del Web scraping
6. Testing enfocado

# RECAPITULANDO...

---

1. Conceptos de una Base de Datos
2. Web Scraping y las Bases de Datos ofuscadas
3. Headless Web Scraping
4. Selenium
5. Defensa ante Web scraping
  1. Riesgos del Web scraping
6. Testing enfocado

Algunas opciones profesionales de web scraping

- Scrapy
- BrightData

# CRÉDITOS

---

## LIBRO RECOMENDADO

<https://www.amazon.com/-/es/Fernando-Rosa/dp/8409363801>

undo de los datos que se dirige utilizarlos en su día a día, como a general interesados en aplicarlos o bien analiza de qué modo los datos

El libro en cuatro grandes bloques: los básicos como dato y algoritmo, temas de inteligencia artificial o de

era muy didáctica el camino para leer y comunicar con datos, es decir, dado *data literacy* o alfabetización

o las empresas pueden incorporar de un método muy sencillo, cómo la empresarial.

tabla de la localización de los datos de la seguridad y de la privacidad. Años, hablaremos, trabajaremos

una sociedad *datificada* que nos a la generación de datos. **DATA** es sin duda

ín

ra los clientes de Adam. tenta.

Fernando de la Rosa  
@titonet

Una edición especial publicada para los clientes de Adam

Fernando de la Rosa  
@titonet



Cómo los datos te ayudarán en tu vida y en tu empresa,  
y transformarán la sociedad

DA  
TA

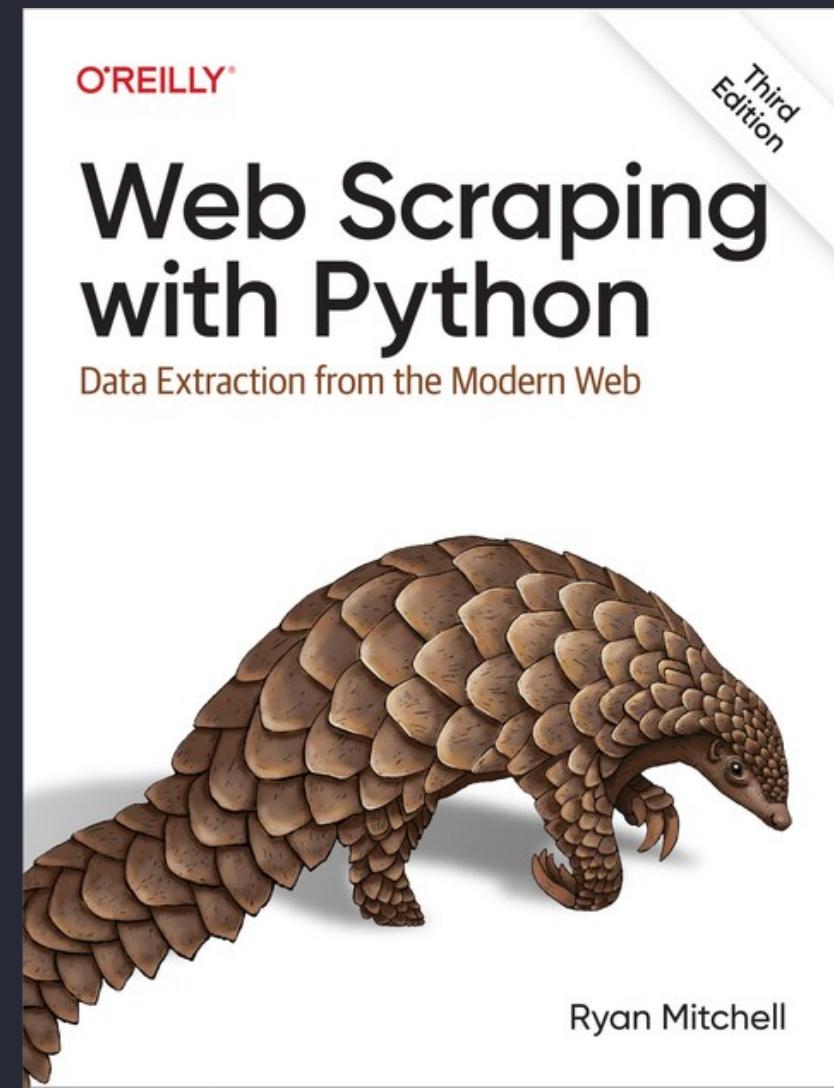
Prólogo de José Mejías y David Ribalta

Company Confidential © Capgemini 2025. All rights reserved | 114



## LIBRO RECOMENDADO

<https://www.amazon.es/web-scraping-python-extraction-modern/dp/1098145356>





## LIBRO RECOMENDADO

[https://openaccess.uoc.edu/bitstream/10609/147437/1/webscraping\\_modulo1\\_webscraping.pdf](https://openaccess.uoc.edu/bitstream/10609/147437/1/webscraping_modulo1_webscraping.pdf)

## ***Web scraping***

PID\_00256970

Laia Subirats Maté  
Mireia Calvo González

Tiempo mínimo de dedicación recomendado: 5 horas



# BIBLIOGRAFÍA

---

Pirámide del conocimiento - DATA: cómo los datos te ayudarán... - Fernando de la Rosa

Códigos HTTP - <https://umbraco.com/knowledge-base/http-status-codes/#:~:text=The%20100%20Continue%20status%20code,the%20request%20has%20already%20finished.>

Unit Testing - <https://www.freecodecamp.org/news/java-unit-testing/>

Repo y notebooks - <https://github.com/jofaval/talks-about/tree/master/knowledge-sharing/web-scraping>



## PREGUNTAS

---

¡¡GRACIAS!!



## About Capgemini

Capgemini is a global leader in partnering with companies to transform and manage their business by harnessing the power of technology. The Group is guided everyday by its purpose of unleashing human energy through technology for an inclusive and sustainable future. It is a responsible and diverse organization of 270,000 team members in nearly 50 countries. With its strong 50 year heritage and deep industry expertise, Capgemini is trusted by its clients to address the entire breadth of their business needs, from strategy and design to operations, fuelled by the fast evolving and innovative world of cloud, data, AI, connectivity, software, digital engineering and platforms. The Group reported in 2020 global revenues of €16 billion.



Get the Future You Want | [www.capgemini.com](http://www.capgemini.com)

This presentation contains information that may be privileged or confidential and  
is the property of the Capgemini Group.

Copyright © 2025 Capgemini. All rights reserved.