Web Scraping

Y LAS BASES DE DATOS OFUSCADAS

Qué aprenderás hoy

Web Scraping

- Normal
- Headless

Selenium

Mundo de los datos

Datos ofuscados

Equipo

A rellenar

Bases de Datos… ¿ofuscadas?

Qué es una Base de Datos

Cuál es su propósito

Conceptos de Bases de Datos

Esquema (DDL)

Ingesta (DML)

Consulta (DML)

Para refrescar

- 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

Por qué ofuscarían los datos las empresas

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

Mostrar información a usuarios, y entorpecérsela a competidores

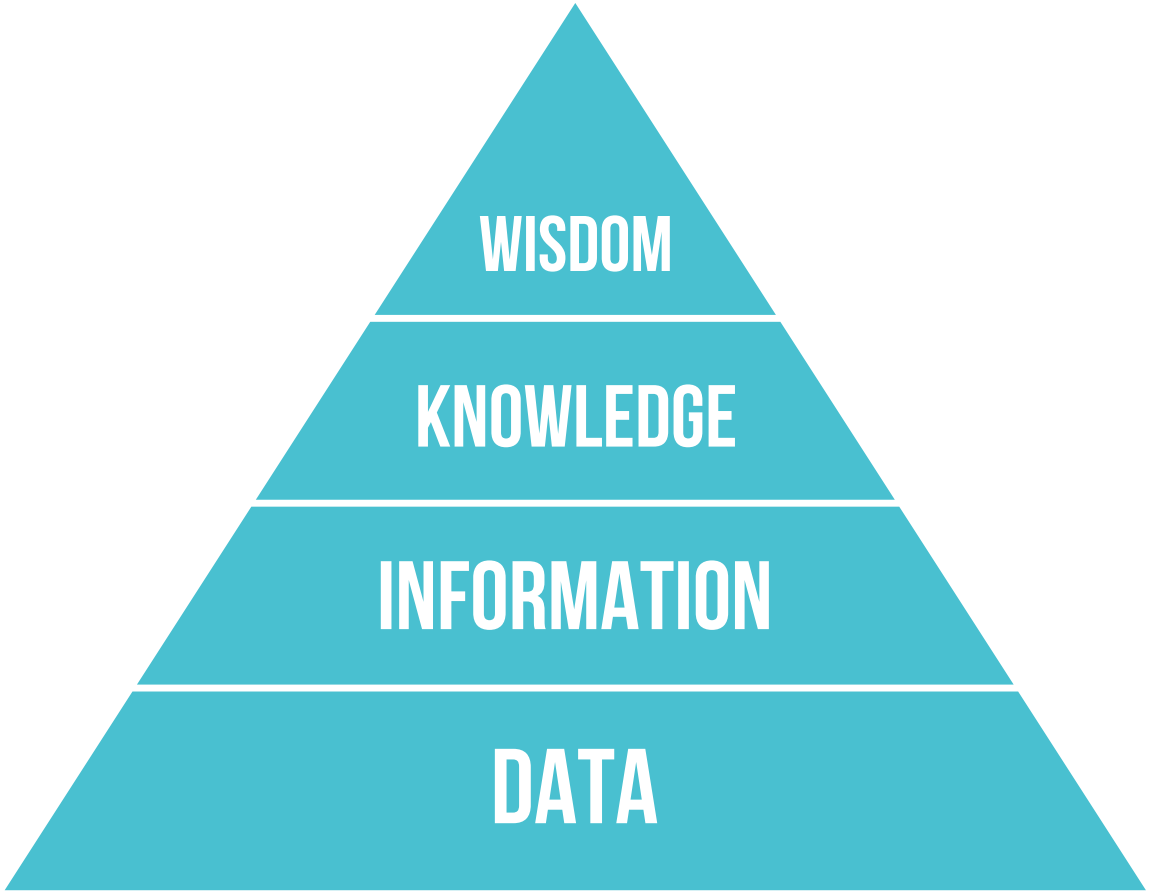
Para qué necesitamos datos

Análisis de Datos

Machine Learning

Seguimiento de ofertas

Comprender mejor un dominio



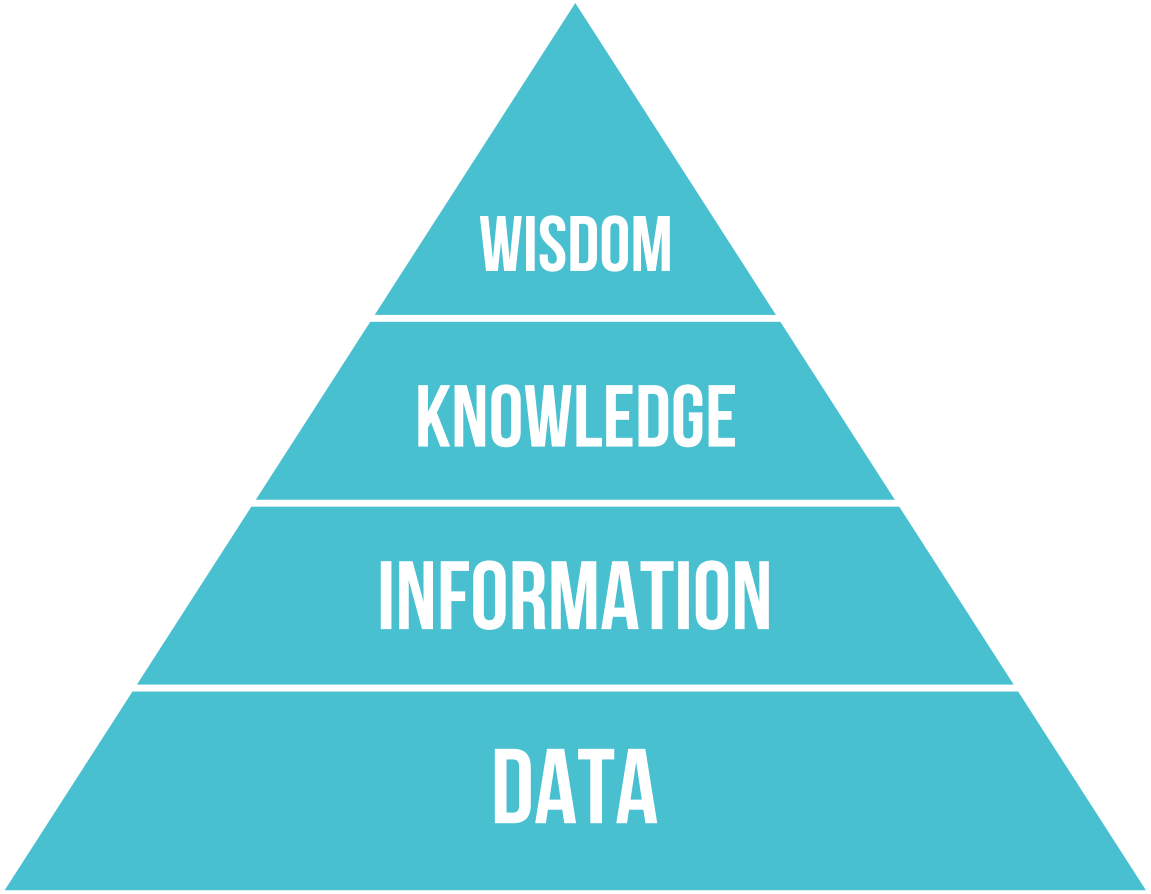
Pirámide del conocimiento

Dato

Información

Conocimiento

Sabiduría



Pirámide del conocimiento

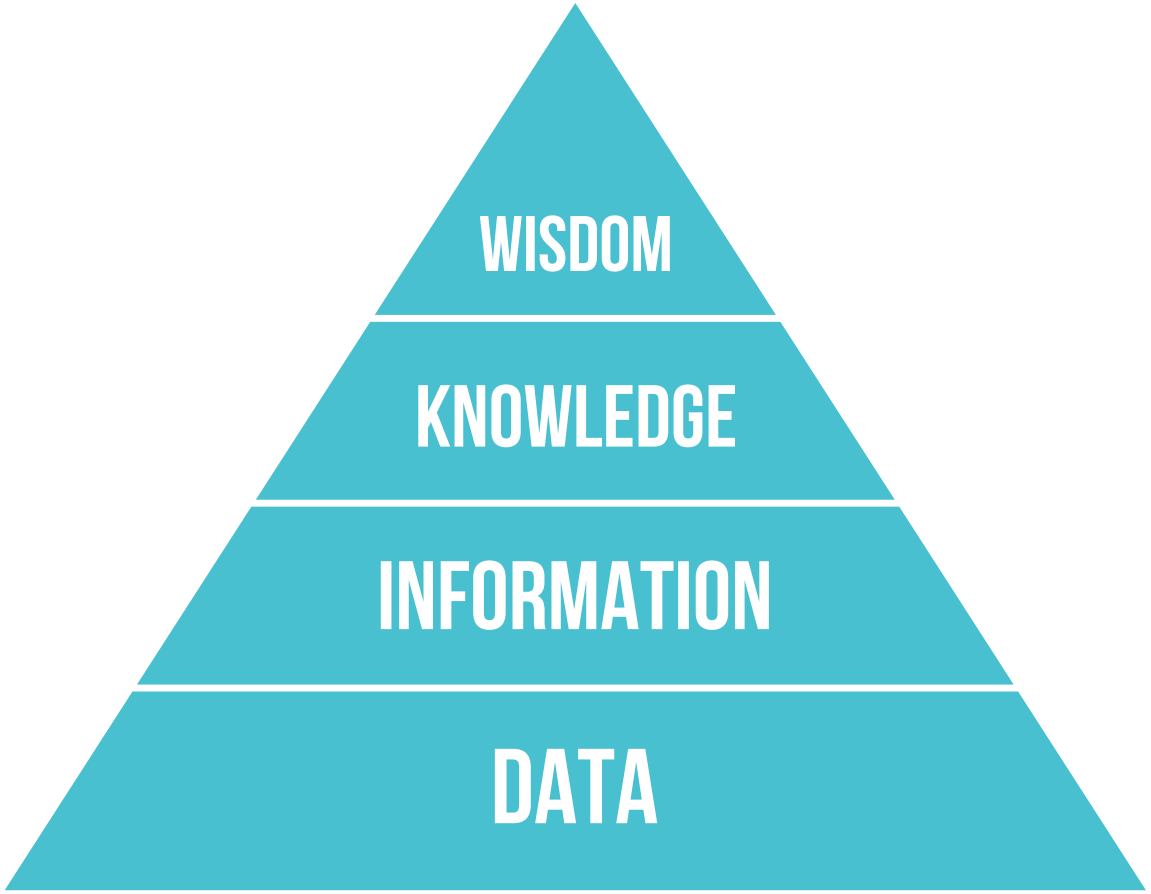
Dato

- Valor puro

Información

Conocimiento

Sabiduría



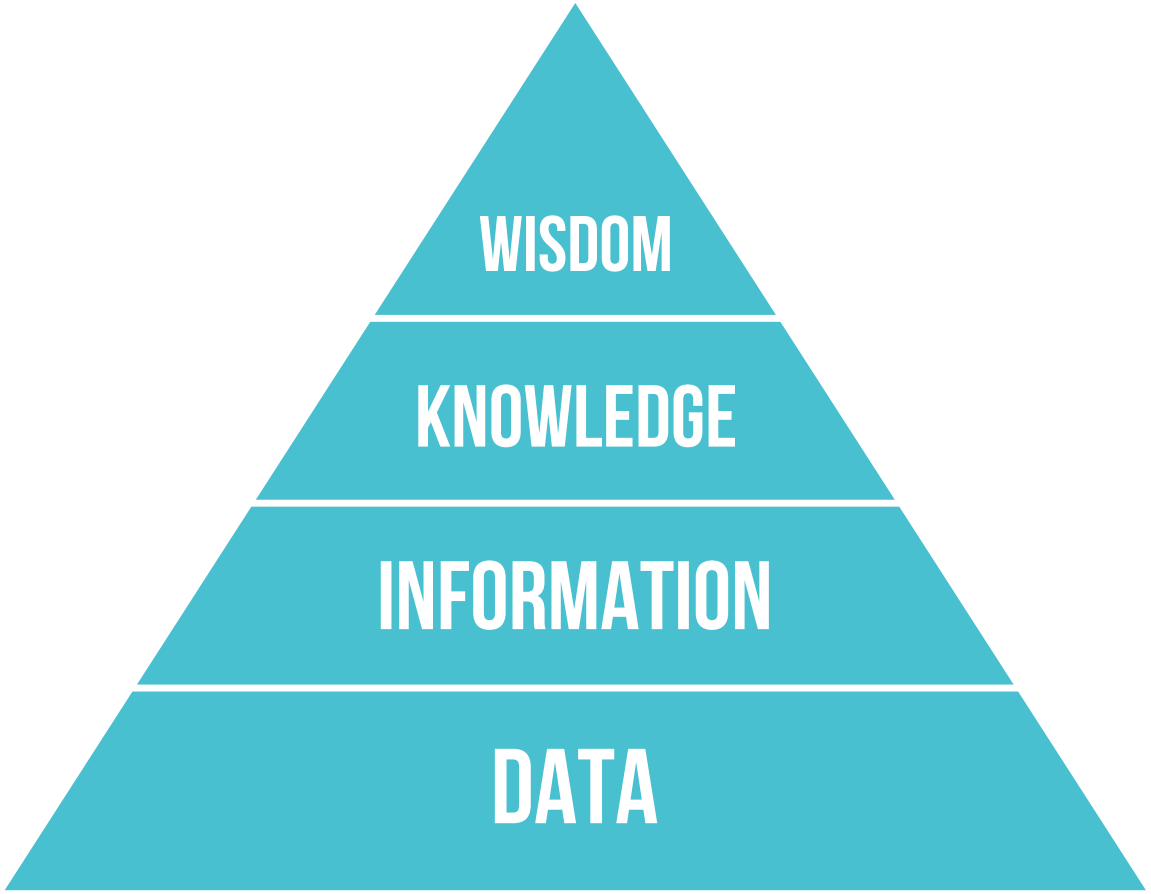
Pirámide del conocimiento

Dato
- Valor puro

Información
- Dato con contexto

Conocimiento

Sabiduría



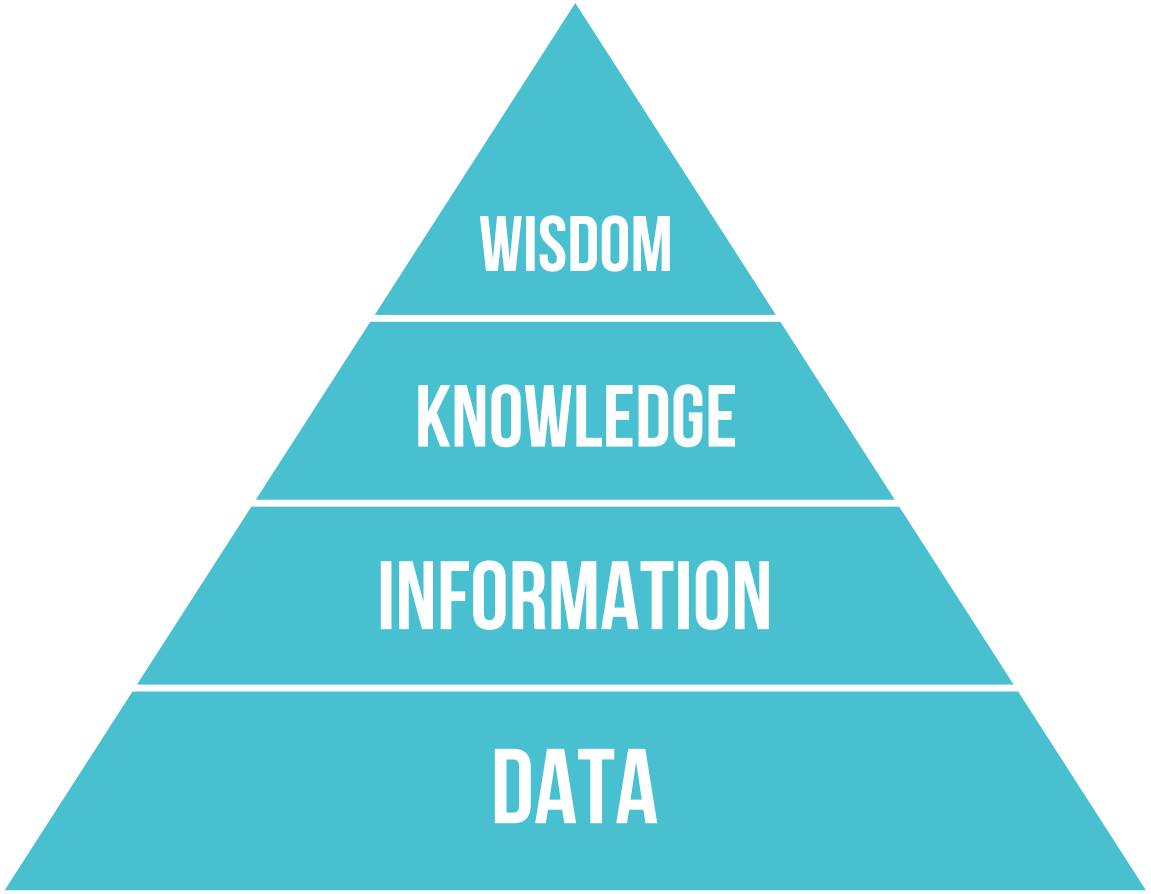
Pirámide del conocimiento

Dato
- Valor puro

Información
- Dato con contexto

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

Sabiduría



Pirámide del conocimiento

Dato
- Valor puro

Información
- Dato con contexto

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

Sabiduría
- Conjunto de conocimiento
- Predicción de comportamientos aquí

Web Scraping

Qué es web scraping

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

Pipeline

1. Request
2. Parser
3. Extract
4. [Iterate]

Pipeline

1. Request
    1. Utilizando el **User Agent** correspondiente, solicitamos el contenido estático
2. Parser
3. Extract
4. [Iterate]

Pipeline

1. Request
    1. Utilizando el **User Agent** correspondiente, solicitamos el contenido estático
2. Parser
    1. Parseamos el contenido utilizando XML, HTML, LXML, a elección
3. Extract
4. [Iterate]

Pipeline

1. Request
    1. Utilizando el **User Agent** correspondiente, solicitamos el contenido estático
2. Parser
    1. Parseamos el contenido utilizando XML, HTML, LXML, a elección
3. Extract
    1. Realizamos queries para extraer la información relevante de la página
4. [Iterate]

Pipeline

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

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

Navergando 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

xPath

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

10 minutos

¿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

DDoS protection (Cloudfare por ejemplo, delays)

Captcha (¿eres un robot?)

Ofuscamiento de classNames (nuestra *guía* cambiante)

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 (Cloudfare por ejemplo, delays)

Captcha (¿eres un robot?)

Ofuscamiento de classNames (nuestra *guía* cambiante)

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 (Cloudfare por ejemplo, delays)

- Delay en la información y registro de sesión

Captcha (¿eres un robot?)

Ofuscamiento de classNames (nuestra *guía* cambiante)

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 (Cloudfare 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)

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 (Cloudfare 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 (Cloudfare 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 Scraping sin Headless

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

5 minutos

Headless Web Scraping

Qué es un Headless Browser

Ejemplos

- Chromium (el engine)
- Selenium

Qué nos ofrece

Conexión ininterrumpida

Simular ser un usuario

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

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

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

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

DEMO Headless Web Scraping

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

10 minutos

Testing

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

¿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?

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

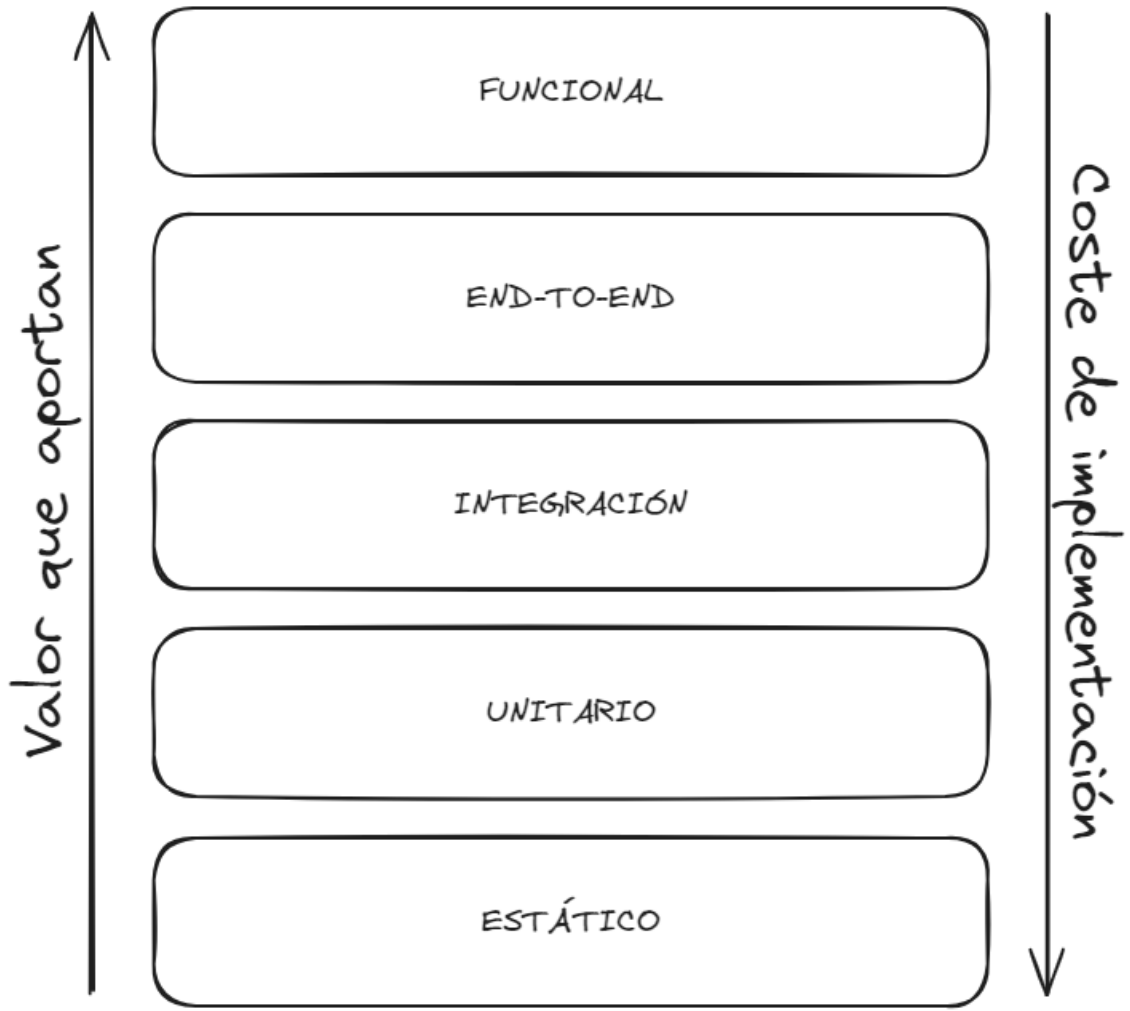
Unitarios
- JUnit, Mockito, Vitest, Enzyme

Integración
- JUnit, Testing Library, Vitest, Enzyme

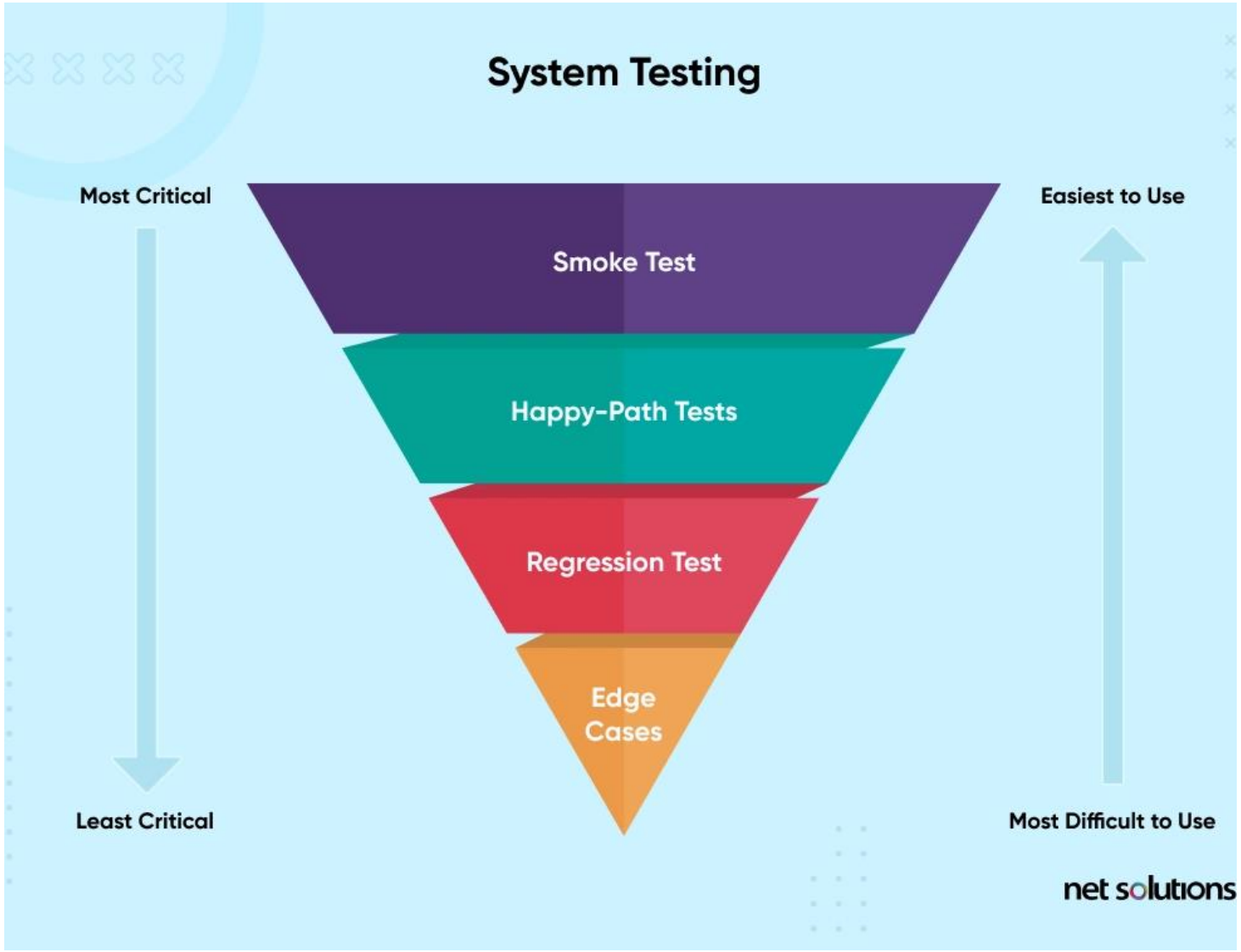
End-to-end (e2e)
- Selenium, Cypress, PlayWright, Puppeteer, Postman

Smoke (Comprobaciones de sistema)
- ping, Kubernetes + Istio

…y unos cuantos más,



Jerarquía de testing

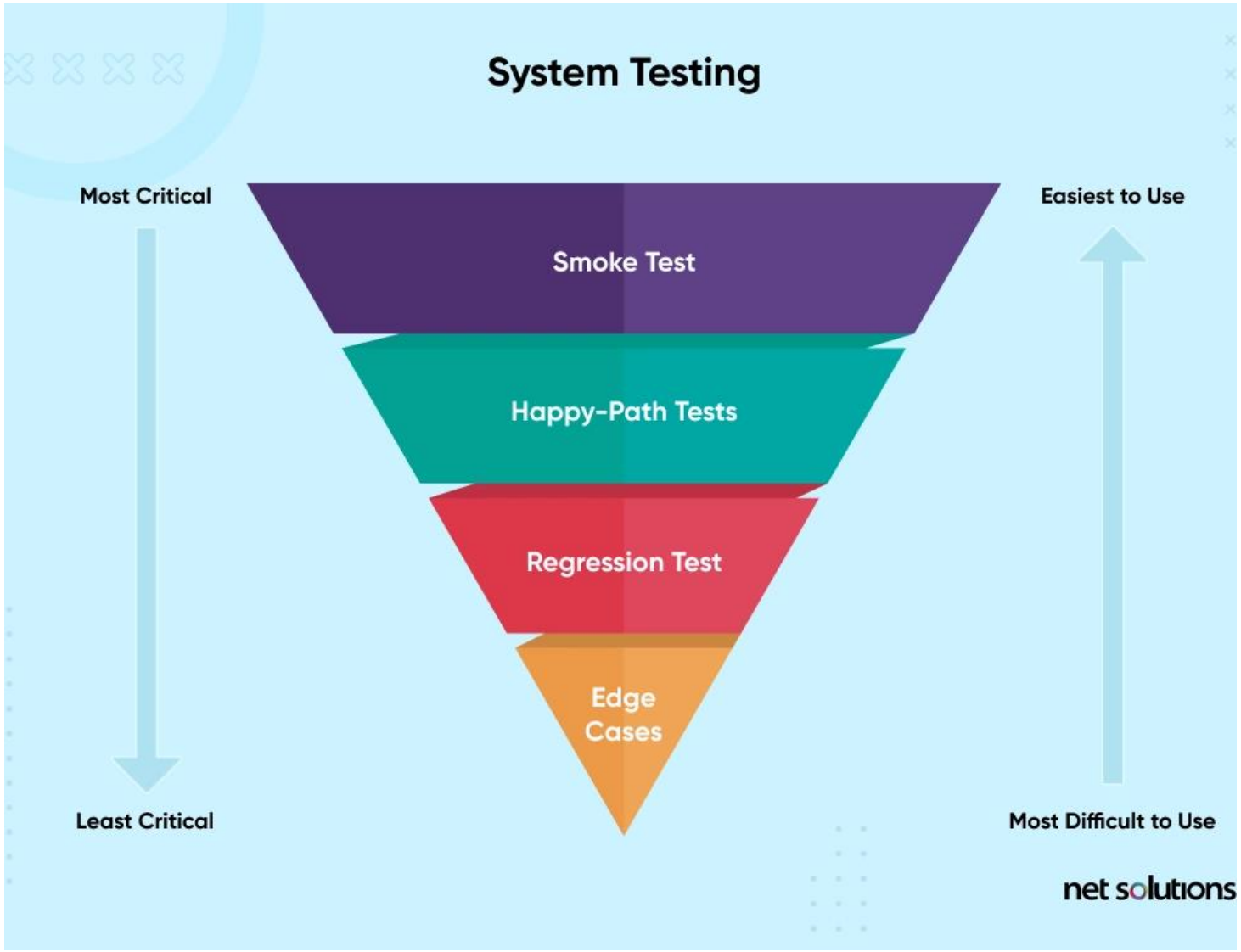


End-to-end

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

Mantenimiento de los tests

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

Recapitulando…

Qué hemos visto

Recapitulando…

1. Conceptos de una Bases de Datos

Recapitulando…

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

Recapitulando…

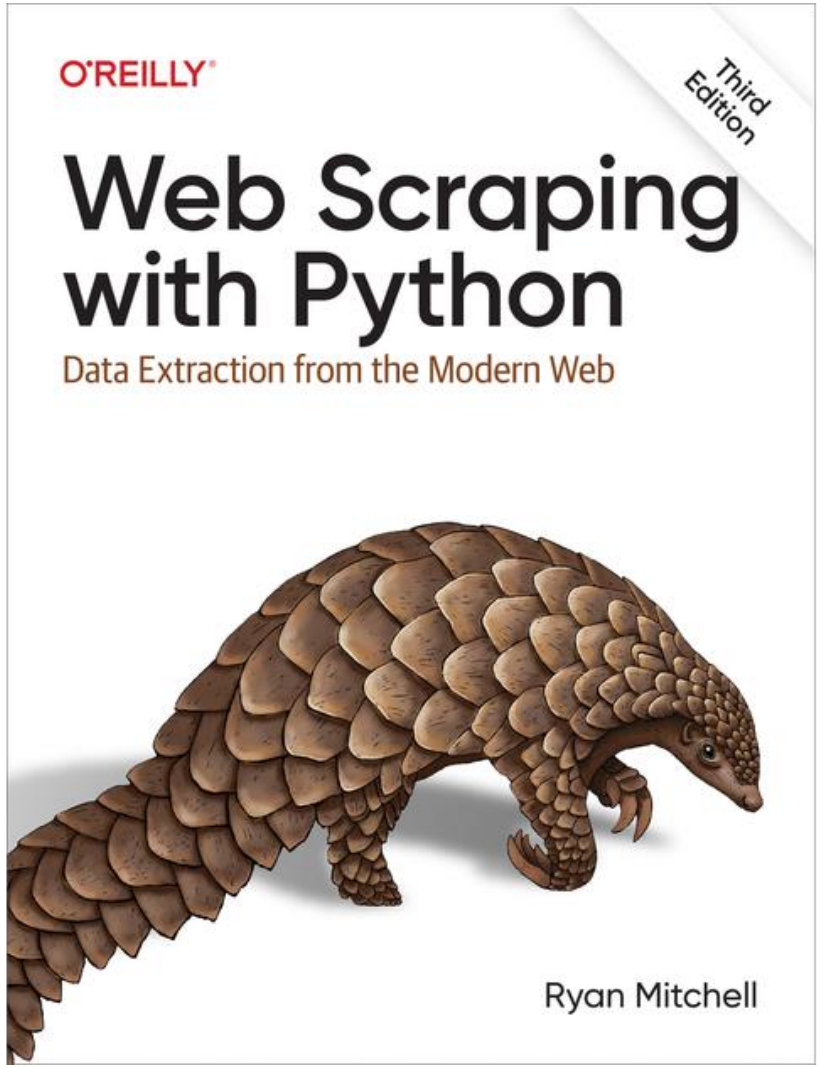
1. Conceptos de una Base de Datos
2. Web Scraping y las Bases de Datos ofuscadas
3. Headless Web Scraping

Recapitulando…

1. Conceptos de una Bases de Datos
2. Web Scraping y las Bases de Datos ofuscadas
3. Headless Web Scraping
4. Opciones profesionales de web scraping
    1. Scrapy
    2. BrightData

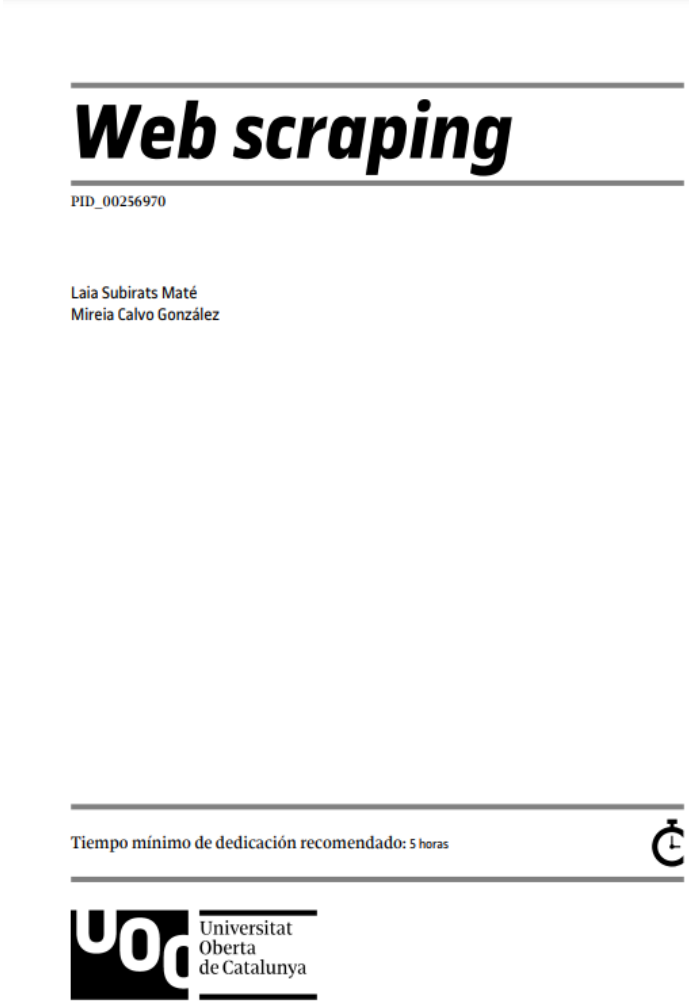
Libro recomendado

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



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

Bibliografía

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.

https://www.freecodecamp.org/news/java-unit-testing/

Diagramas, labs y más contenido - https://github.com/jofaval/talks-about/tree/master/uv/web-scraping-y-las-bases-de-datos-ofuscadas