

# Obtención de datos de Internet a través de R y otros métodos.

## Iniciación

# Introducción

# Introducción

## Información del docente

## Perfil del participante

## Objetivos

¿Guarda este curso relación con el Big Data?

¿Qué es el web scraping?



Antonio Manuel Moreno Moreno  
**tonimoreno@us.es**

Profesor de Sistemas de Información para la Empresa  
Departamento de Economía Financiera y Dirección de Operaciones  
Facultad de Ciencias Económicas y Empresariales  
Universidad de Sevilla

Líneas de investigación:

- Industria 4.0
- Impacto de las redes sociales en el Turismo mediante análisis de reviews
- Fake reviews (Deep learning)
- crowdfounding para PYMES

Fundador y Responsable tecnológico de :



Start-Up de la US (primer premio 2020) dedicada a la evaluación del encaje cultural en organizaciones

# Perfil del participante:

No se requiere un nivel mínimo de conocimiento de R, ni de ningún lenguaje de programación.

Partimos desde cero.

Orientado principalmente aunque no exclusivamente a :

Investigadores de cualquier rama de las ciencias sociales interesados en aprovechar posibilidades de estas herramientas para obtener datos para sus propias líneas de investigación.

Alumnos de doctorado

# Objetivos:

Conocer qué posibilidades existen para obtener información desde documentos en **diversos formatos** y/o publicada en **Internet** (web, RSS, Excel, csv, pdf)

Obtener de **forma semiautomática** bases de datos (tablas)

Conocer herramientas de **software gratuitas**  
**Open Source – Código Abierto**



# ¿Guarda este curso relación con el Big Data?

El concepto de Big Data es muy amplio. Habitualmente hace referencia a toda una serie de procesos y acciones encaminadas a la captura, almacenamiento y posterior análisis de grandes cantidades de información.

“El término ‘big data’ se refiere a los datos que son tan grandes, rápidos o complejos que es difícil o imposible procesarlos con los métodos tradicionales. **El acto de acceder y almacenar grandes cantidades de información para la analítica ha existido desde hace mucho tiempo.** Pero el concepto de big data cobró impulso a **principios de la década de 2000** cuando el analista de la industria, Doug Laney, articuló la definición actual en torno a las 3 Vs: **Volumen, Velocidad y Variedad.**”

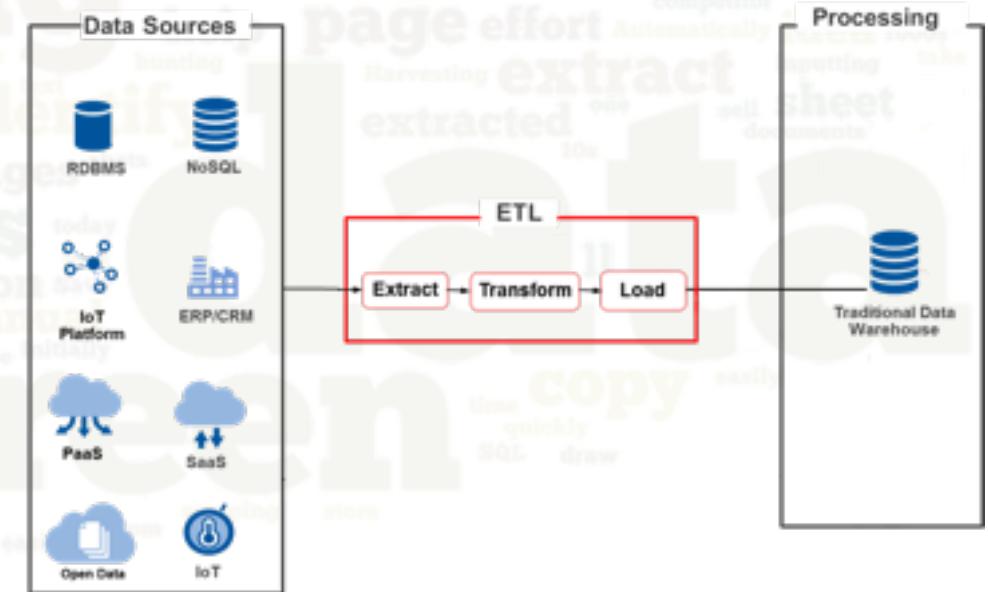
[https://www.sas.com/es\\_es/insights/big-data/what-is-big-data.html](https://www.sas.com/es_es/insights/big-data/what-is-big-data.html)

# ¿Guarda este curso relación con el Big Data?

Por tanto, en sentido estricto este curso no versa sobre Big Data, aunque algunas de las técnicas y procedimientos que se utilizan son los mismos con la diferencia de la escala de operaciones.

El Big Data abarca desde el acceso a las fuentes de datos, extracción, transformación, procesamiento, análisis, y presentación de la información.

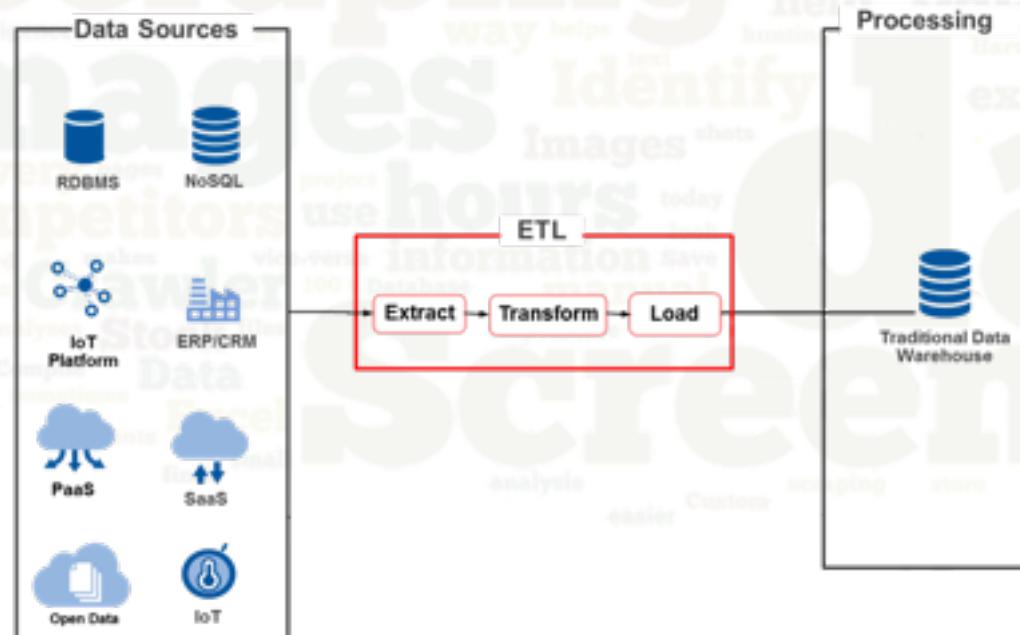
Todo ello orientado a mejorar la toma de decisiones en las empresas o la extensión de las fronteras del conocimiento en el ámbito académico.



# ¿Guarda este curso relación con el Big Data?

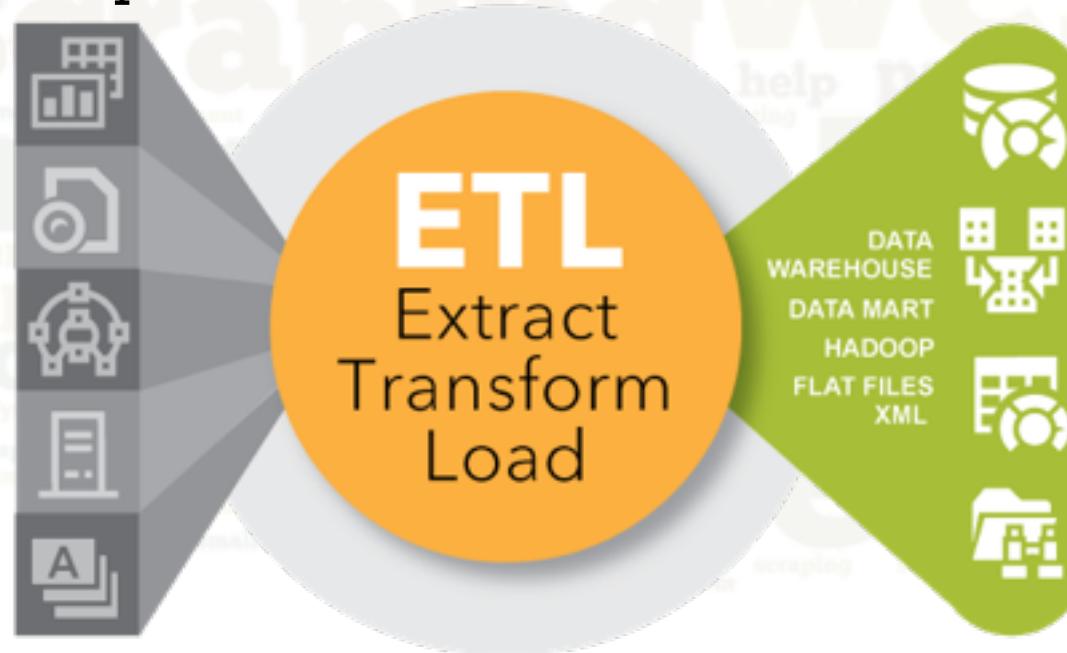
Los procedimientos y técnicas que veremos en este curso forman parte de la **fase ETL**, dentro del esquema típico del Big Data.

**ETL = EXTRACT + TRANSFORM + LOAD**

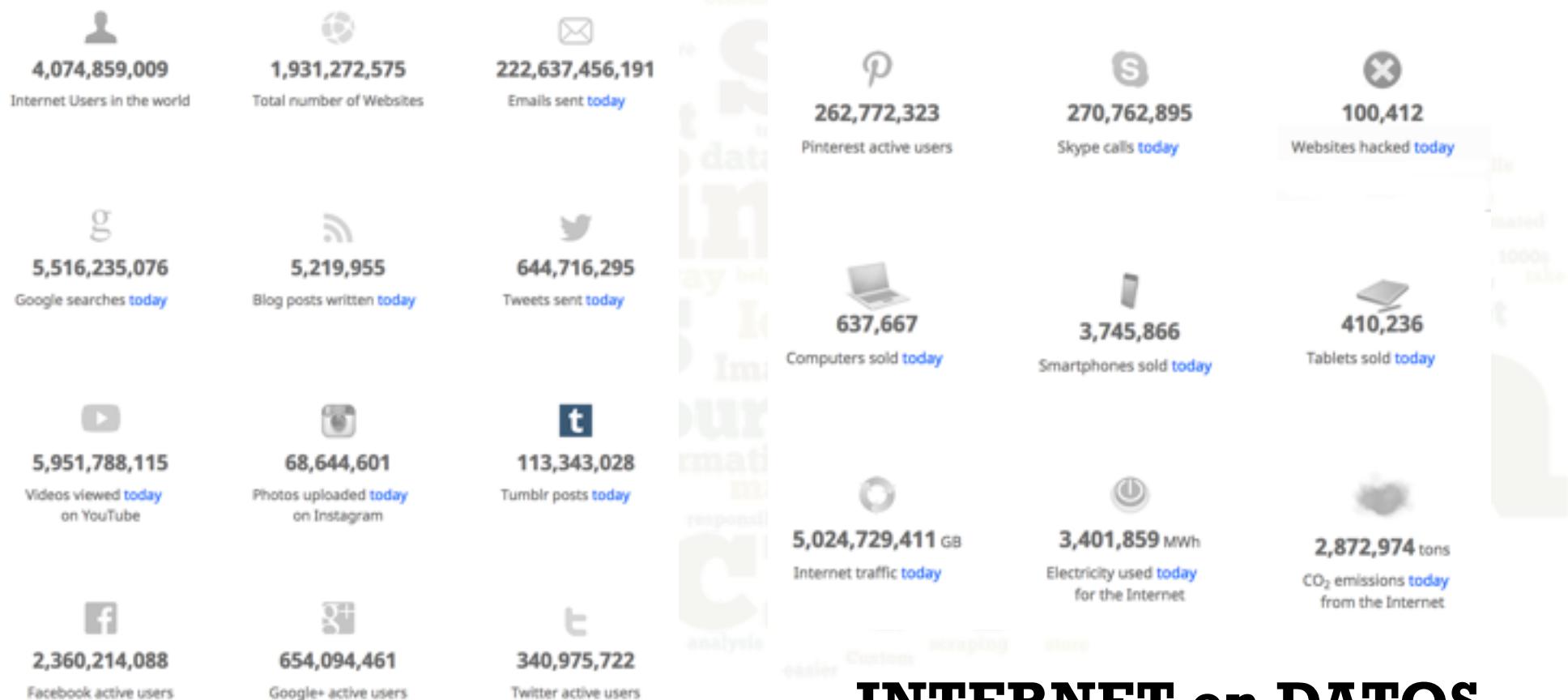


# ¿Guarda este curso relación con el Big Data?

**Extract, Transform , Load (ETL)** es un proceso de **integración de datos** que transfiere datos brutos desde diversas fuentes a una base de datos alojada en un servidor destino, preparando/estructurando la información para usarla en análisis posteriores.



# ¿Guarda este curso relación con el Big Data?



# ¿Guarda este curso relación con el Big Data?

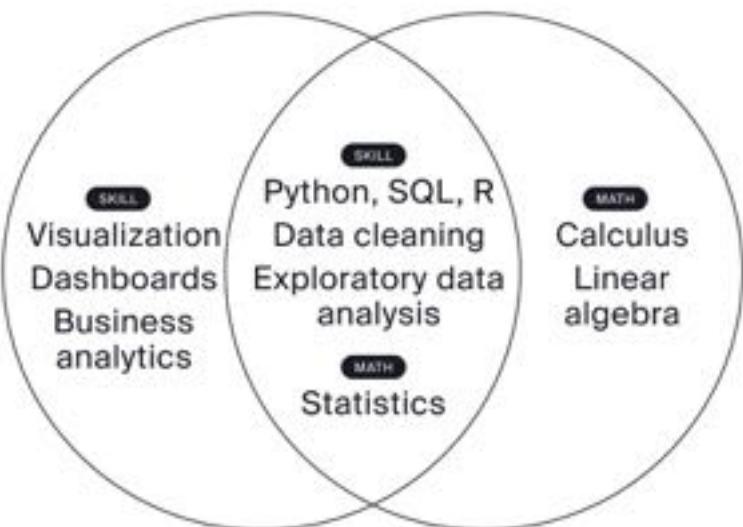


**INTERNET en DATOS**

By the way, in the 203 seconds approximately 4582522 GB of data was transferred over the internet.

# ¿Guarda este curso relación con el Big Data?

## Hard skills: Data Analysis vs. Data Science



Aunque las profesiones alrededor de los datos se han ido especializando. Todas ellas requieren el dominio de una serie de competencias comunes en mayor o menor medida.

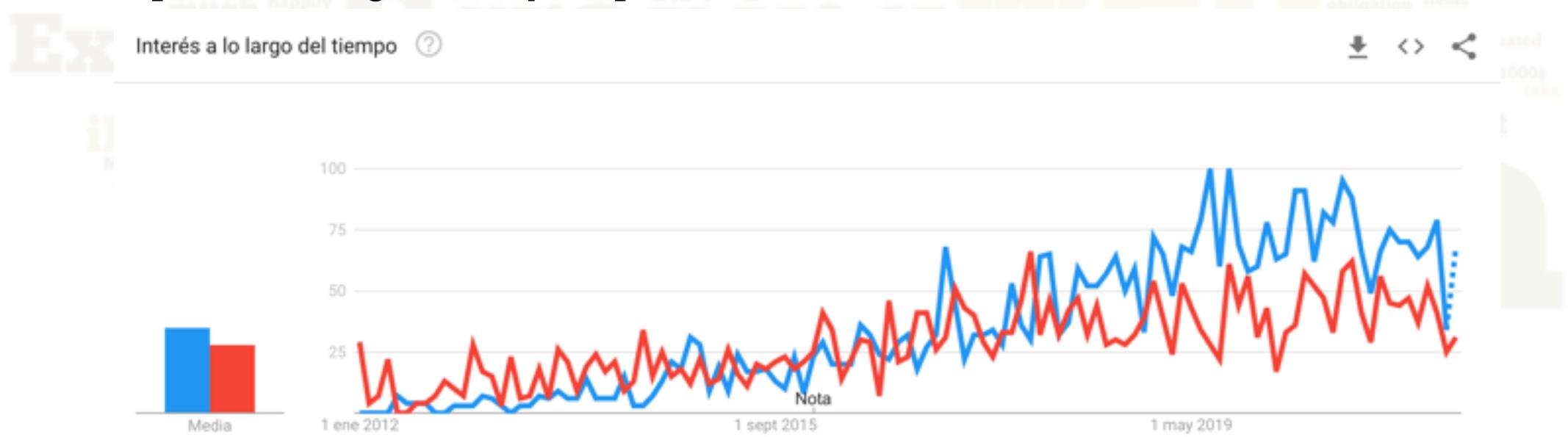
Fuente: <https://medium.com/practicum-by-yandex/data-analysis-or-data-science-4c690a468c5d>

# ¿Qué es el Web Scraping?

El **web scraping** es un **conjunto de técnicas o procedimientos que sirven para extraer información de páginas web de forma automatizada**.

# ¿Qué es el Web Scraping?

Con el incremento de contenidos disponibles en Internet el **web scraping** se convierte en una posibilidad cada vez más interesante para investigadores y empresas.



Fuente: Google Trends

En azul: data science; en rojo: web scraping

<https://trends.google.es/trends/explore?date=today%205-y&geo=ES&q=data%20science,%2Fm%2F07ykbs>

# ¿Qué es el Web Scraping?

Para realizar web scraping se pueden utilizar multitud de aplicaciones y procedimientos diferentes.

Para elegir el procedimiento más adecuado (el menos costoso o menos complejo) en cada caso, será útil aprender algunas cuestiones sobre cómo se muestran los datos en internet.

## Bloque 1

**¿Cómo se muestran los datos en Internet?**  
Problemática de su obtención  
en forma de base de datos.

En Internet podemos encontrar una gran variedad de fuentes de información.  
(en diferentes formatos)

En este curso nos centraremos principalmente en el formato más característico de INTERNET y que conforma la World Wide Web.

Páginas WEB

**HTML + CSS**



## **HTML (Hypertext Markup Language)**

Lenguaje de Etiquetas indica al navegador las **partes del contenido**. Y cómo mostrarlas.

Un archivo escrito en lenguaje HTML es un **fichero de texto** que contiene etiquetas para organizar los contenidos.

## **CSS (Cascading Style Sheets)**

Define **cómo** se muestran los elementos de un archivo HTML.

## **XHTML = XML + HTML**

XHTML es una redefinición de HTML con sintaxis XML. Es una versión más “ limpia ” y más consistente de HTML.

## **SOLO veremos HTML**

## **Tecnologías WEB**

HTML	SQL
XHTML	ASP
CSS	ADO
XML	PHP
JavaScript	.NET
VBSCRIPT	SMIL
DOM	SVG
DHTML	FLASH
AJAX	Java applets
E4X	Java servlets
WMLScript	Java Server Page

HTML es desarrollado inicialmente por Tim Berners-Lee en el CERN (Centro Europeo de Investigación Nuclear) en 1980. Posteriormente se han desarrollado multitud de tecnologías que han hecho evolucionar a la WWW (nacida el 12 de marzo de 1989) hasta lo que es hoy.

## **Algunos ejemplos de documentos XML (existen miles ...)**

- \* Las facturas electrónicas.
- \* RSS Feed de Noticias (probar en Chrome y Firefox)

<https://servicios.elpais.com/rss/> (directorio)

<http://ep00.epimg.net/rss/elpais/portada.xml>

- \* Los documentos de WORD, EXCEL, ... se guardan en un formato derivado del XML.

Existen muchos “dialectos” del XML, nosotros mismos podemos crear uno con un determinado propósito. Ej: Biblioteca / Estantes / Nivel / Libro

## Ejemplo de código HTML

```
<html>
  <head>
    <title> Titulo de pagina </title>
  </head>
  <body>
    <h1> Primera pagina. </h1>
    <p> Esta es mi primera pagina.
      Y dentro de ella mi primer parrafo<p>
      <b> Este texto está en negrita </b>
    </body>
</html>
```

Un documento XML/HTML tiene una estructura en forma de árbol, como puede observarse. Las etiquetas guardan una jerarquía.

Los navegadores (FIREFOX, CHROME, Internet Explorer, Safari, Opera, etc.) al interpretar los documentos XML/HTML crean una estructura lógica denominada **DOM (Document Object Model)** que funciona como una API (application programming interface).

Esta API puede ser utilizada para consultar y/o modificar la representación de la página que hace el navegador .

## Otras Etiquetas importantes:

```
<div> </div> <ul><li></li> </ul> <table><tr><td></td><td></td><td></td></tr></table>
```

```

```

```
<a href="http://www.us.es">Universidad de Sevilla</a>
```

# DOM (Document Object Model)

Estructura de datos en forma de árbol que el navegador web genera internamente, a la que se puede acceder mediante la **consola** o **snippets** dentro de las **Herramientas para Desarrolladores**. Es utilizada por los programadores web para introducir efectos dinámicos en las páginas. En Scraping **se puede utilizar para identificar y extraer contenidos**.

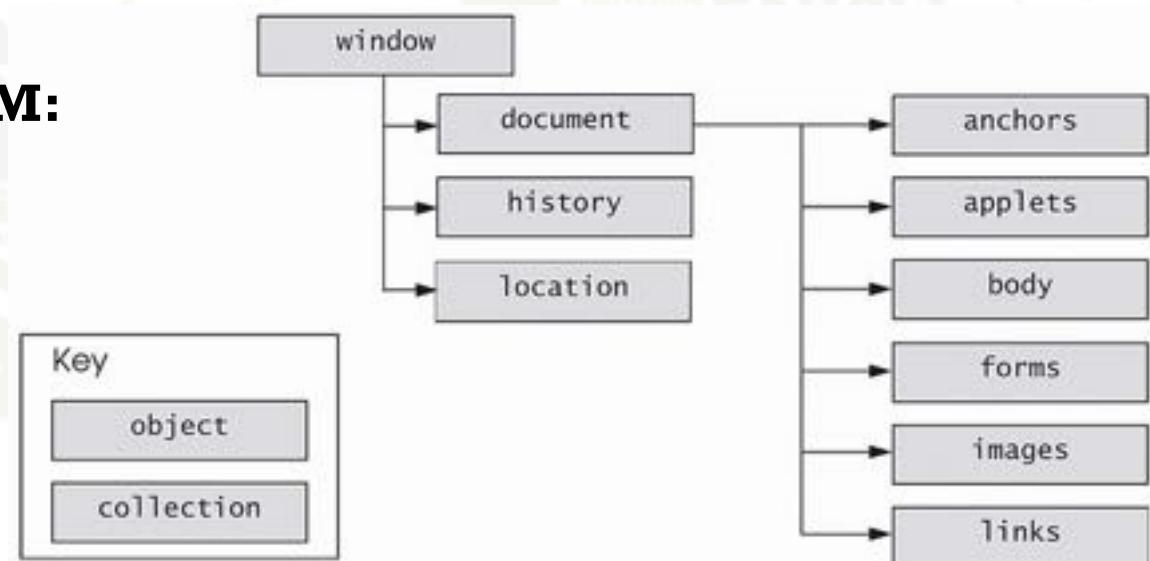
## Piezas importantes del DOM:

Nodos / Elementos

Atributos

Propiedades

Métodos



## Bloque 1 – Parte 2

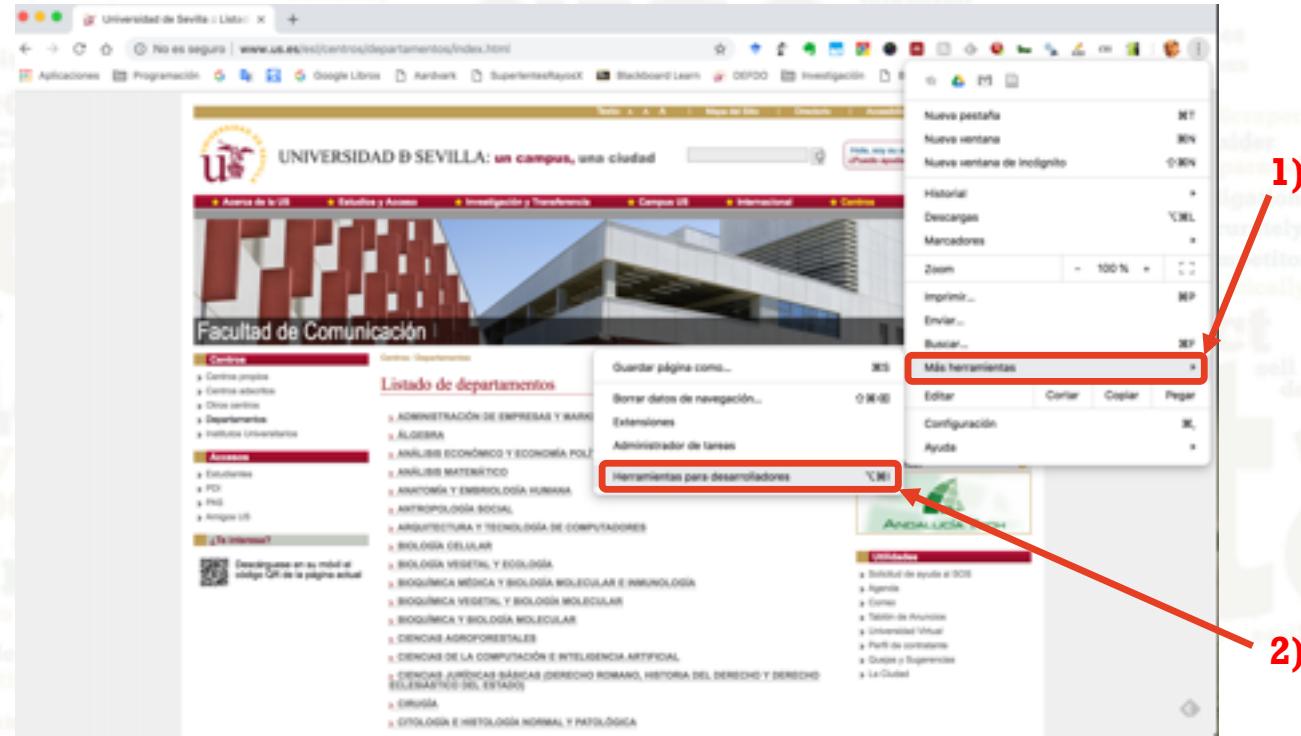
Herramientas para la obtención  
de datos en páginas web.  
Iniciación y Aplicación práctica.

# Cómo identificar contenidos en páginas web

## Ejemplo captura desde consola navegador

# Ejemplo de cómo extraer información directamente del HTML I

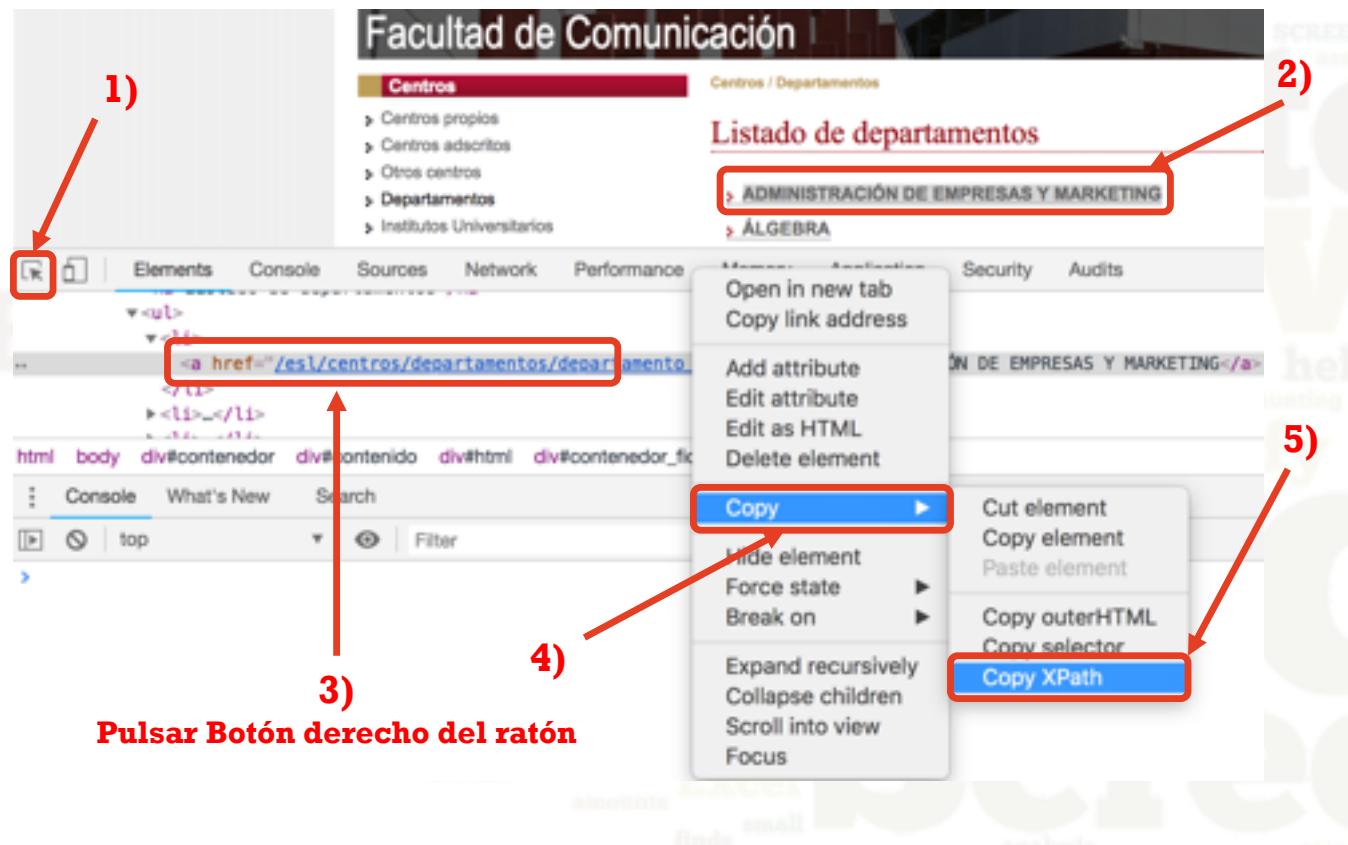
Listado de Departamentos US.ES : <https://www.us.es/centros/departamentos>



Abrimos en el **Navegador CHROME** las **Herramientas para Desarrolladores** (Developer Tools)

Abrir en : W10 = **F12**, Ctrl + Mayús + I    Mac = Cmd + Opción + I

## Ejemplo de cómo extraer información directamente del HTML II



Pulsar Botón derecho del ratón

**XPath** = XML Path Language.  
Lenguaje que usa "path like" sintaxis para **identificar y seleccionar nodos** en un documento XML.

Con ella podemos acceder al elemento seleccionado y a todos sus atributos a través del **DOM**

Una vez terminado el paso 5, tenemos la ruta XPath del elemento seleccionado.  
Con ella podemos acceder al elemento y a todos sus atributos a través del DOM

## Ejemplo de cómo extraer información directamente del HTML III

```
//*[@id="contenedor_ficha"]/ul/li[1]/a
```

(1) Identificador XPath del elemento seleccionado en paso 2)

```
$x('//*[@id="contenedor_ficha"]/ul/li[1]/a')
```

(2) Instrucción que obtiene un conjunto de nodos del DOM a través de su identificador XPath. Se ejecuta en la Consola de las Dev Tools.

CUIDADO con las comillas. Vease como la expresión (1) aparece dentro de unas comillas simples. Ej: '(1)'

```
$x('//*[@id="contenedor_ficha"]/ul/li/a')[18].textContent
```

Modificamos la instrucción anterior para obtener el texto del elemento número 19.  
**RECUERDE que el contador empieza en 0.**

```
$x('//*[@id="contenedor_ficha"]/ul/li/a')[1].href
```

En este caso queremos obtener el "link" del elemento número 2

## XPath:

Expresión	Descripción
nodename	Selecciona todos los nodos con el nombre "nodename" podría ser : div, table, span, tr, td, h1 ... etc
/	Selecciona desde el nodo raiz
//	Selecciona nodos en el documento desde el actual, que coincidan con la selección no importa donde esten
.	Selecciona el nodo actual
..	Selecciona el nodo padre de el actual
@	Selecciona atributos

Ejemplo: `//*[@id='titleCast']//span[@class='itemprop']`

Esta expresión significa : Buscamos cualquier etiqueta “span” con el atributo `class='itemprop'` no importa lo que sea PERO localizada bajo una etiqueta `div[@id='titleCast']`

## Ejemplo de cómo extraer información directamente del HTML IV

### Snippet :

Seleccione pestaña : Sources > snippets

The screenshot shows the Chrome DevTools interface with the Network tab selected. A red arrow labeled '1)' points to the 'Sources' tab in the top navigation bar. Another red arrow labeled '2)' points to the 'Snippets' section in the left sidebar. The main content area displays a block of JavaScript code:

```
1 cont = "";
2 contenido = $x('//*[@id="contenedor_ficha"]/ul/li/a');
3 for(var i=0; i < contenido.length; i++){
4     cont = cont + "|" + contenido[i].href;
5 }
```

The code is annotated with line numbers 1 through 5. The code's purpose is to extract href values from all links within a specific ul element with id="contenedor\_ficha".

### Snippet 1:

```
cntnd = $x('//*[@id="contenedor_ficha"]/ul/li/a');
for(var i=0; i < cntnd.length; i++){
    console.log(cntnd[i].textContent);
}
```

### Snippet 2:

```
cont = "";
cntnd = $x('//*[@id="contenedor_ficha"]/ul/li/a');
for(var i=0; i < cntnd.length; i++){
    cont = cont + "|" + cntnd[i].href;
}
```

## Explicación del Javascript utilizado :

```
for(var i=0; i < f; i++){ <instrucciones> }
```

Bucle que hace tomar a la variable ‘i’ valores desde 0 hasta ‘f’, repitiendo sucesivamente las <instrucciones> que contiene.

```
console.log( variable );
```

Muestra en la consola una determinada “variable”

```
cont = cont + " | " + cntnd[i].href;
```

Carga en la variable “cont” una expresión (marcada en verde).

## Bloque 1 – Parte 3

Utilidad del WebCrawling y WebScraping  
estático y dinámico. Ejemplos.

**“The open web is by far the greatest global repository for human knowledge, there is almost no information that you can't find through extracting web data. This list of tools will help you take advantage of this information for your own projects and businesses.  
Happy scraping!”**

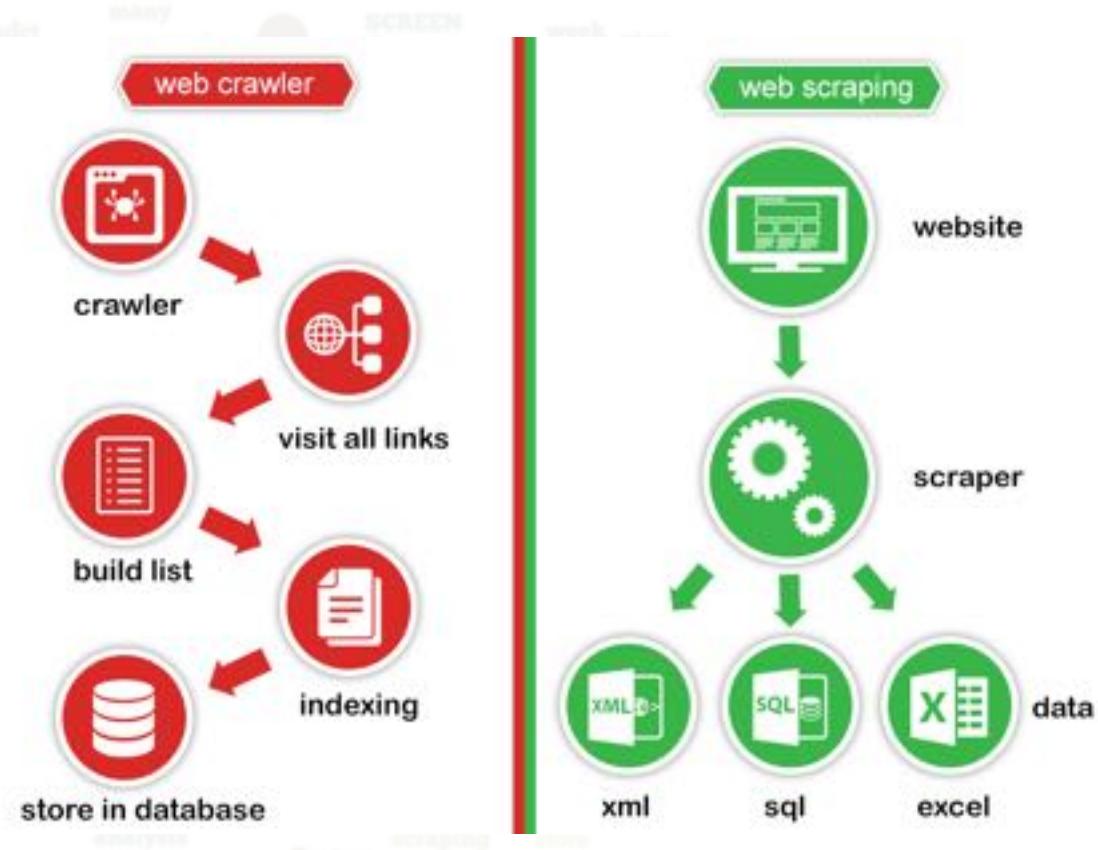
Fuente: <https://www.scrapingapi.com/blog/the-10-best-web-scraping-tools>

# Extracción semiautomática de datos de Internet

Es necesario utilizar un **web crawler** cuando los contenidos que necesitamos extraer (**web scraping**) están distribuidos en una multitud de páginas web.

Normalmente se utiliza una base de datos con las URLs (direcciones en Internet) como índice. Esa base de datos podemos crearla desde una página donde aparezca una lista.

El crawler recorrerá el índice obteniendo las páginas contenidas en él. Una vez obtenidas esas páginas se les aplicará un algoritmo de scraping para extraer los datos que se pretenden conseguir.



Nota: En el **snippet 2** se puede ver una forma de crear un índice.

## Dificultades scraping:

- Detección uso abusivo y bloqueo IP desde servidor.
- Utilización masiva de Javascript
- Cambios frecuentes de estructura/estilo website
- Limitación de acceso por CAPTCHA
- Limitación acceso a determinados dispositivos (agents)
- Carga de contenidos a partir de acciones del usuario  
(pulsar botón, o desplazarse al final de la página)

## Efecto del Javascript en una página web :

Ejemplos: <https://www.rtve.es/> , Google.com o Elpais.com

Para deshabilitar Javascript de forma permanente en Chrome, tendremos que dirigirnos a :

Menú -> Configuración > Privacidad y seguridad > Configuración de sitios -> Contenido > **Javascript**

y lo deshabilitamos.

Podemos llegar a este panel de opciones rápidamente si escribimos en la barra de direcciones : **chrome://settings/content/javascript**