



# Almacenamiento y captura de datos

Claudio Aracena

GobLab - Universidad Adolfo Ibáñez  
Chatbot Chile



# Contenidos

- Captura de datos desde archivos
- Base de datos
- Captura y almacenamiento de datos en BD
- **Captura de datos de la Web (Web scraping)**
- Captura de datos de API (ej: Twitter)
- Captura y almacenamiento en arquitecturas Big data

Códigos y clase en:

<https://github.com/caracena/almacenamiento-captura-datos>

# Clase de hoy



## Captura de datos de la Web (Web scraping)

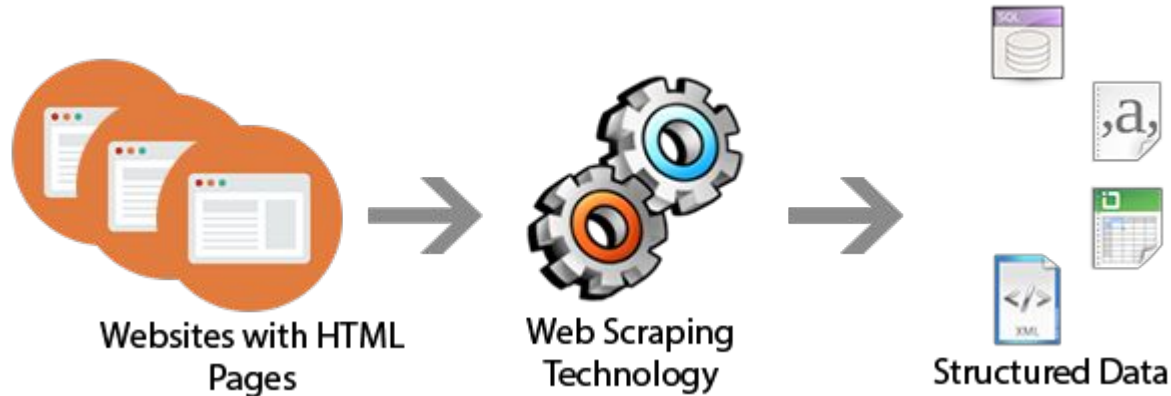
- Web scraping
- Librerías de web scraping en Python
- Web scraping simulando navegación





# Web scraping

Web scraping es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.



# HTML



HTML es el lenguaje de marcado que se utiliza en la mayoría de los sitios webs. Es similar a XML, ya que cuenta con elementos con tags que contienen data.

A screenshot of a web browser interface. The left pane shows the HTML source code, and the right pane shows the rendered page. The code includes a DOCTYPE declaration, a title tag, a heading tag, and a paragraph tag. The rendered page displays the heading and paragraph, along with a URL at the bottom.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Result Size: 358 x 358

## This is a Heading

This is a paragraph.

<https://www.w3schools.com/>



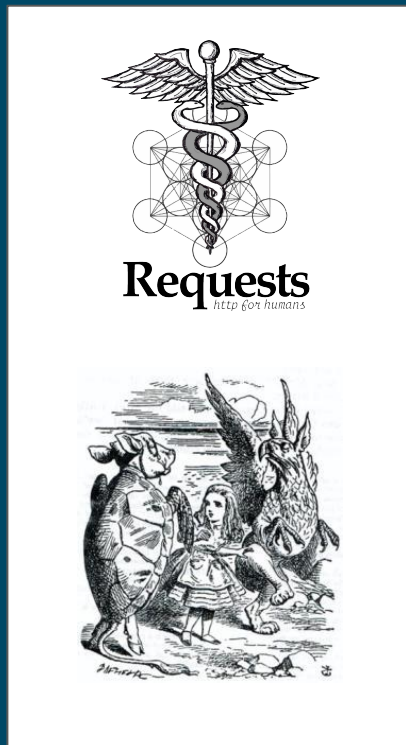
# Librerías de web scraping en Python

## Librerías de requests:

- requests
- urllib.request

## Librerías de parseo

- BeautifulSoup
- lxml



# Web scraping simulando navegación



Muchos sitios no pueden ser accedidos desde una url o utilizan intensivamente Javascript para sus funcionalidades. Para los web scrapers, esto significa que no pueden obtener la información necesaria de una manera tradicional.

Para superar estas barreras se emula un navegador y luego se extrae la data requerida. La herramienta número uno para automatizar la navegación en un browser es Selenium.





# Consideraciones éticas

- Si hay disponible una API, utilizarla para evitar raspar toda una página
- Consultar datos a una tasa de tiempo razonable (incluir un delay)
- Sólo retener información que me es necesaria y que no puedo conseguir por otros medios
- Respetar otros términos establecidos en sitios Web
- Respetar derechos de autor