

I M T 1 0 0 1

I N T R O D U C C I Ó N A I N G E N I E R Í A M A T E M Á T I C A

2 0 2 3 - 1

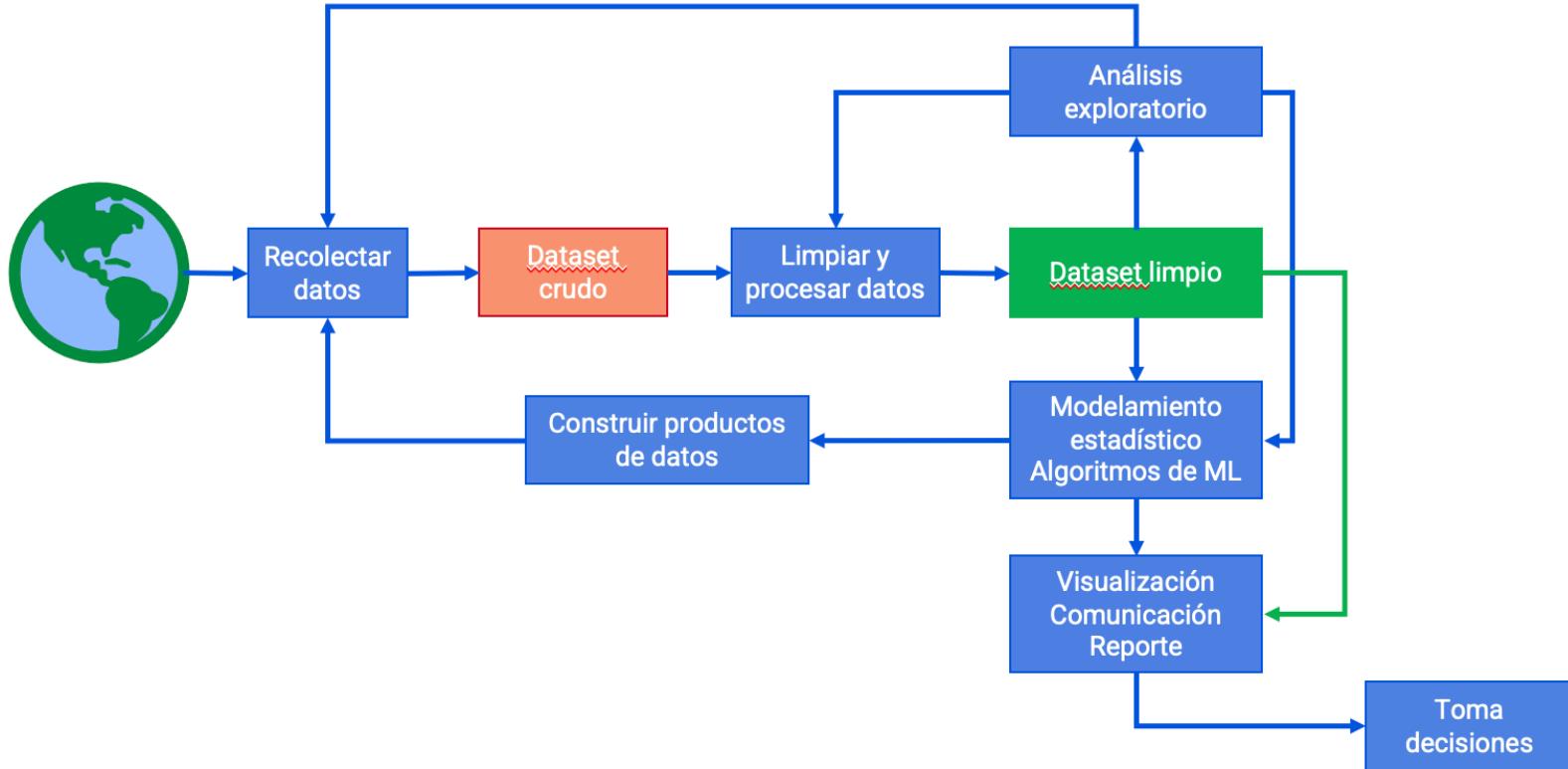
EXTRACCIÓN Y TRANSFORMACIÓN DE DATOS Y TEXTO

PAULA AGUIRRE – TALLER 4

ESTRUCTURA DEL TALLER

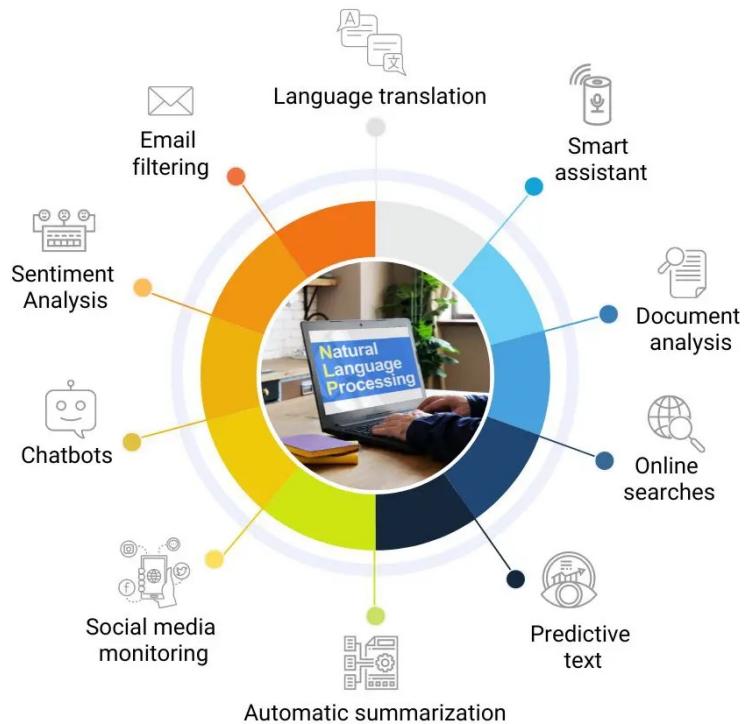
1. Motivación: NLP y datasets de entrenamiento
2. Extracción de datos desde la web
3. Procesamiento y análisis de datasets de texto
4. Precauciones y limitaciones de datos extraídos desde la web.

PROCESO DE CIENCIA DE DATOS



PROCESAMIENTO DE LENGUAJE NATURAL

- Proceso mediante el cual los computadores entienden y procesan el lenguaje humano natural.



@megynkelly
wants Sad beat #AmericaFirst #VoteTrump show
Republican country won Crooked President
Enjoy Hillary just many Sanders interviewed
look jobs soon Cruz people Join way back Caroline
Conveniently made like I'm day Bernie interviewed
time Trump New get bad want State
another AMERICA go vote amp much totally believe
news Ted even Rubio @realDonaldTrump falling
tonight now U watch win media amazing
must #Trump2016 one good Wow
#MakeAmericaGreatAgain speech himWithYou
Party @CNN big @FoxNews money
tomorrow love campaign
Pennsylvania dishonest Donald
total Indiana



Robust Detection of Extreme Events Using Twitter:
Worldwide Earthquake Monitoring

Barbara Poblete, Jheser Guzmán, Jazmine Maldonado and Felipe Tobar

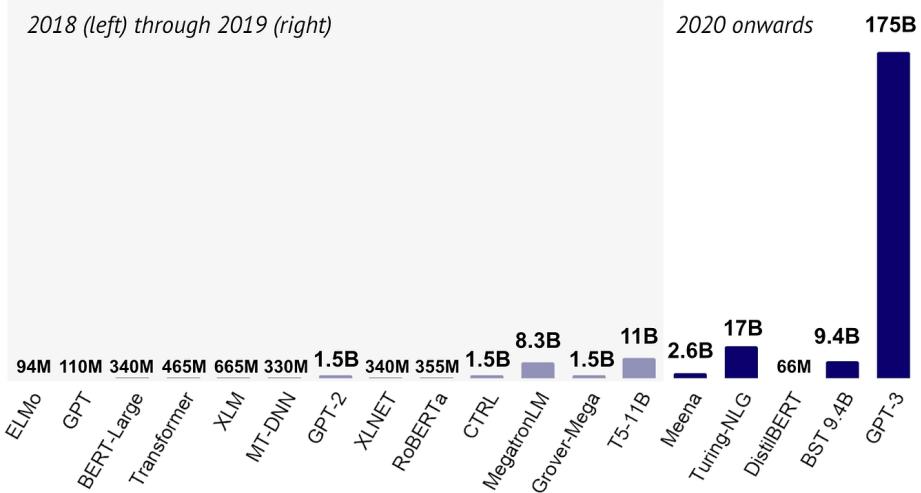
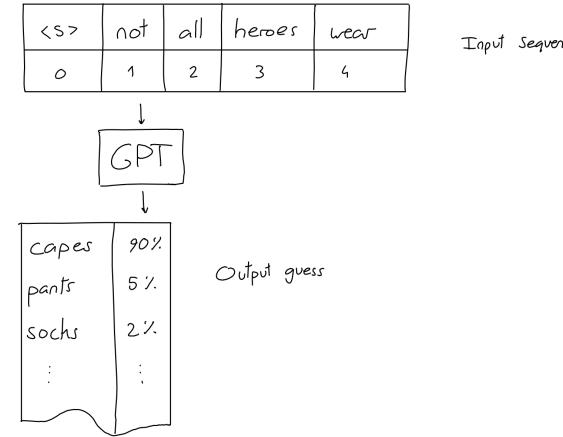


MODELOS DE LENGUAJE

Modelos que determinan la **probabilidad de una secuencia de palabras** en una frase, en un lenguaje en particular, en base al análisis de cuerpos de texto.

Large Language Models (LLMs) ➔ modelos de lenguaje de gran complejidad

- basados en algoritmos de aprendizaje profundo (redes neuronales) con grandes cantidades de parámetros
- entrenados en base a datasets de texto masivos.
- Pueden capturar patrones complejos de lenguaje y generar texto casi indistinguible del generado por humanos.



CHAT GPT

<https://openai.com/blog/chatgpt>

M Tell me how ChatGPT works.

ChatGPT is a large language model that uses deep learning techniques to generate human-like text. It is based on the GPT (Generative Pre-trained Transformer) architecture, which uses a transformer neural network to process and generate text. The model is pre-trained on a massive dataset of text, such as books, articles, and websites, so it can understand the patterns and structure of natural language. When given a prompt or a starting point, the model uses this pre-trained knowledge to generate text that continues the given input in a coherent and natural way.

GPT : Generative Pre-trained Transformer

- *Generative* ➔ predice la siguiente palabra
- *Pre-trained* ➔ previamente entrenado sobre grandes cantidades de datos
- *Transformer* ➔ red neuronal basada en encoder-decoder

Introducing ChatGPT

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

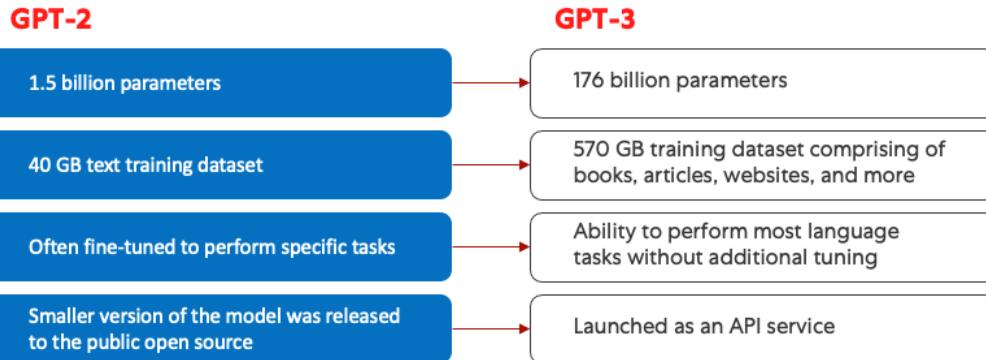
[Try ChatGPT ↗](#)

[Read about ChatGPT Plus](#)

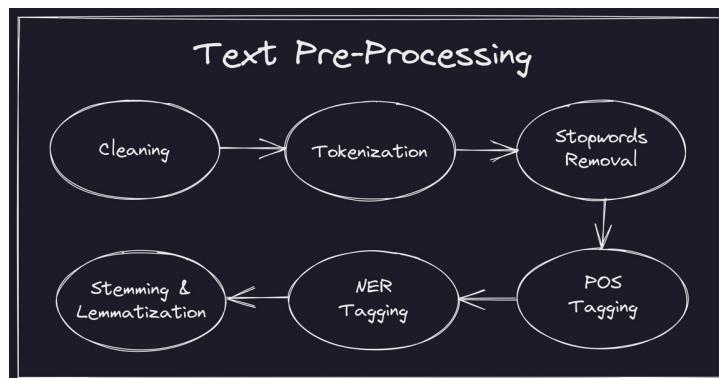
CHAT GPT

Datos de entrenamiento (GPT-3): WebText

- Dataset masivo de textos de internet : 570 GB, 300 billones de palabras.
- Diversas fuentes: libros, Wikipedia, artículos, redes sociales, documentos legales, etc.
- Varios pasos de procesamiento, anotación y transformación de datos.



<https://paperswithcode.com/dataset/webtext>



¿Dónde encontramos datos, y cómo
los obtenemos?

Fuente de datos

Primarias

Datos obtenidos directamente de la fuente

Secundarias

Datos previamente recolectados

Internas

Datos provenientes de la propia organización

Externas

Datos relativos a otras personas u organizaciones

Privadas

Datos de acceso limitado a ciertos usuarios autorizados

Abiertas

Datos accesibles en forma libre y gratuita

FUENTES DE DATOS

¿Dónde podemos encontrar datos?

- Documentos, archivos y sistemas de información privados.
- Bases de datos en ubicaciones específicas.



Data en la web

- **Abiertos:** <https://www.freecodecamp.org/news/https-medium-freecodecamp-org-best-free-open-data-sources-anyone-can-use-a65b514b0f2d/>
- **Propietarios:** data que pertenece a, y es administrada por, un individuo, organización o grupo
- Ejemplos:



DATOS ABIERTOS

- Datos disponibles en **forma gratuita y sin restricciones** de derechos de autor (copyright), patentes u otros mecanismos de control
- Pueden ser **utilizados, reutilizados y redistribuidos** libremente por cualquier persona
- Sujetos, cuando más, al requerimiento **de atribución** y de **compartirse** de la misma manera en que aparecen, respetando la **seguridad y privacidad** de la información

¿Por qué datos abiertos? → bien público

- **Transparencia**
- Generación de **valor social y comercial**: los datos son un recurso clave para actividades sociales y económicas, y para impulsar negocios y servicios innovadores.
- **Participación y compromiso**: todos podemos acceder y contribuir a los datos e información.



Support a fair, free and open future.

VALUE STORIES

Open data businesses - an oxymoron or a new model?

Building a business based on open data may seem counterintuitive, but new models are emerging with greater frequency and demonstrating how to integrate open data into a business operation in a useful and profitable manner. Identifying the type of open data that can help a business grow involves not only understanding what open data is, but also creative thinking around what can be done with the data. Once a useful data source is identified, a business owner must assess the risks and decide how to integrate the data into their product. While the first part of open data use relies...

[Read More](#)

Danish address registry

In 2002, the Danish government, having determined that "free and unrestricted access to addresses of high quality is beneficial to the public and forms the basis for reaping substantial benefits in public administration and in industry and commerce", released its official Danish address database free of charge. Eight years later, the government analysed the impact of opening up Danish address data and came to the following conclusion. Reuse: In 2010, free-of-charge address data was delivered to total of 1,236 parties of which 70% were from private companies, 20% from the central government and 10% from municipalities. ...

[Read More](#)

Making aid more effective in Nepal

Nepal is currently focusing on building transparent and accountable public institutions following a period of disruptive civil war. By 2013-14, foreign aid represented 22% of the national budget and financed most development spending. NGOs, journalists and civil society have demanded more comprehensive, timely and detailed information on aid flows, particularly geographic information, to show where money is being directed. In June 2013, the Aid Management Platform was launched by the Ministry of Finance to assist efforts aimed at monitoring aid and budget spending. All NGOs are now required to report details about their funding and programmes to the platform, building...

[Read More](#)

<http://opendatahandbook.org/value-stories/en/>

¿Qué tipos de datos podemos encontrar?

TIPOS DE DATOS

Estructurados

- Esquema predefinido y homogéneo.
- Estructura tabular con filas y columnas.
- Fácil de analizar y modelar.

Year,Make,Model,Price
1997,Ford,E350,3000.00
1999,Chevy,"Venture Extended Edition",4900.00
1999,Chevy,"Venture Extended Edition",5000.00
1996,Jeep,Grand Cherokee,4799.00

Semi-estructurados

- No están organizados en filas y columnas.
- Cuentan con llaves y etiquetas que proporcionan una estructura jerárquica a los datos.
- Análisis requiere más trabajo previo.

```
[  
  {  
    "age_adjusted_death_rate": "7.6",  
    "death_rate": "6.2",  
    "deaths": "32",  
    "leading_cause": "Accidents Except Drug Poisoning (V01-X39, X43, X45-X59, Y85-Y86)",  
    "race_ethnicity": "Asian and Pacific Islander",  
    "sex": "F",  
    "year": "2007"  
  },  
  {  
    "age_adjusted_death_rate": "8.1",  
    "death_rate": "8.3",  
    "deaths": "87",  
    ...  
}
```

No estructurados

- No hay un estructura o jerarquía interna clara.
- Existen muchos formatos nativos no estructurados.
- Más difícil de analizar.

Call me Ishmael. Some years ago—never mind how long precisely—having little or no money in my purse, and nothing particular to interest me on shore, I thought

DATOS ESTRUCTURADOS

- Data altamente organizada y fácil de descifrar.
- Está en un formato predefinido (tabla o base de datos)
- Formatos comunes: `.csv`, `.tsv`, `.txt`, `.xlsx`, `SQL`
- **Dato (datum)** → una observación o abstracción de una entidad real (persona, objeto o evento). Puede estar descrita por uno o más atributos.
- **Datos (data/dataset)** → conjunto homogéneos de datos relativo a una colección de entidades.

Year, Make, Model, Price

1997, Ford, E350, 3000.00

1999, Chevy, "Venture Extended Edition", 4900.00

1999, Chevy, "Venture Extended Edition", 5000.00

1996, Jeep, Grand Cherokee, 4799.00

	A	B	C
1	name	age	height
2	Michael	46	5'9"
3	Jim	31	6'0"
4	Pam	29	5'7"
5	Meredith	53	5'6"
6	Dwight	35	5'10"

Name	Dry/Wet Food	Good Boy (Y/N)
Fido	Dry	Y
Rex	Wet	N
Bubbles	Dry	Y
Cujo	Wet	N

Tag #	Height (in)	Weight (lbs)
1573	15	21
2684	9	7
3795	27	130
4806	6	5

Tag #	Name	Breed	Color	Age
1573	Fido	Beagle	Brown/White	1.5
2684	Rex	Pekingese	White	9
3795	Bubbles	Rottweiler	Black	5
4806	Cujo	Chihuahua	Gold	4

DATOS ESTRUCTURADOS

Dataset

Atributos

ID	NOMBRE	APELLIDO	PROMEDIO	INGRESO
1	Luciano	Solari	6.9	2021
2	Andrea	Ramirez	6.2	2022
3	Mario	Diaz	5.8	2021
...

→ Dato/datum/registro/instancia

DATOS SEMI-ESTRUCTURADOS

XML: Extensible Markup Language

- Metalenguaje diseñado para almacenar y transportar datos.
- Diseñado para ser auto-descriptivo.

```
<?xml version="1.0" encoding="UTF-8"?>
<websites>
  <website id="133">
    <title lang="en">File Extension Database</title>
    <url>https://www.file-extension.info</url>
    <category>Data Formats</category>
  </website>
</websites>
```

JSON: JavaScript Object Notation (<https://www.json.org/json-en.html>)

- Formato de texto liviano para intercambio de data, fácil de interpretar por humanos (archivo de texto simple) y de generar y formatear (parse) para máquinas.
- Estándar para envío de data mediante entre servidores y aplicaciones web.
- Estructura auto-descriptiva.
- Semejante a un diccionario de Python, con dos estructuras base:
 - **keys**: strings
 - **valores**:
 - 4 tipos de datos atómicos: number, string, boolean, null
 - 2 tipos de datos compuestos: array, object

```
{
  "departamento": 8,
  "nombredepto": "Ventas",
  "director": "Juan Rodríguez",
  "empleados": [
    {
      "nombre": "Pedro",
      "apellido": "Fernández"
    },
    {
      "nombre": "Jacinto",
      "apellido": "Benavente"
    }
  ]
}
```

DATOS SEMI-ESTRUCTURADOS - JSON

- La estructura puede ser **anidada**: el valor de un atributo es un nuevo diccionario, o conjunto de pares atributo-valor.
- Para grandes conjuntos de datos, permite evitar repeticiones y campos en blanco ➔ formato más liviano
- Existen varias librerías que permiten trabajar con datos en formato json:
- **json**: librería con funciones básicas para leer, escribir y analizar datos en formato JSON.

<https://docs.python.org/3/library/json.html>

`json.loads` ➔ leer un archivo .json

```
{  
    "departamento": 8,  
    "nombredepto": "Ventas",  
    "director": "Juan Rodríguez",  
    "empleados": [  
        {  
            "nombre": "Pedro",  
            "apellido": "Fernández"  
        }, {  
            "nombre": "Jacinto",  
            "apellido": "Benavente"  
        }  
    ]  
}
```



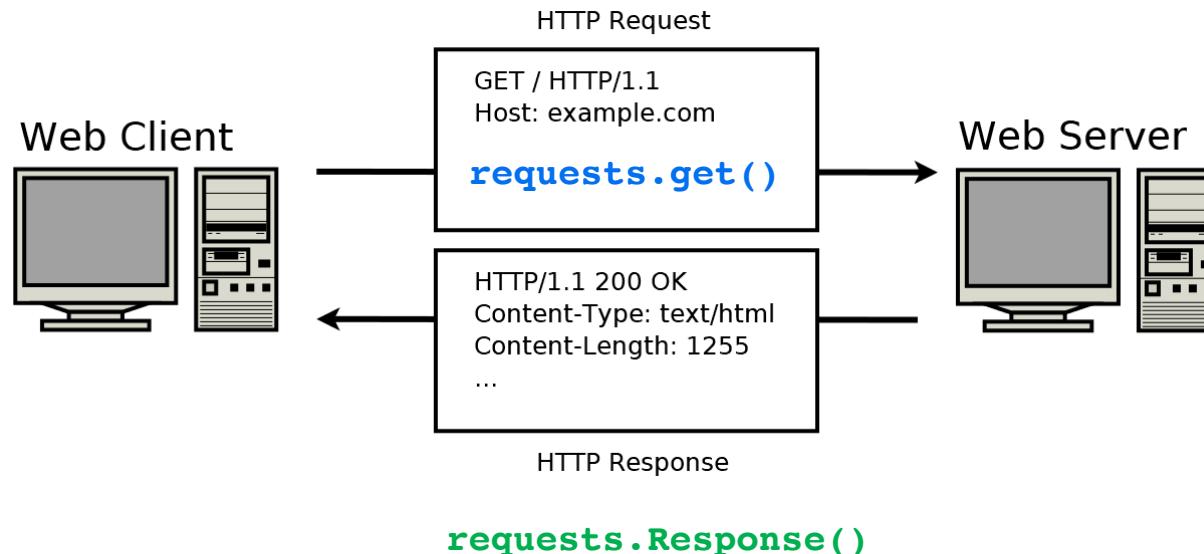
¿Cómo extraemos datos desde la
Web?

EXTRACCIÓN DE DATOS DESDE LA WEB

- Hay 3 formas de extraer datos de la web:
 - **URL:** apertura y descarga de datos a partir de Universal Resources Locator (link)
 - **API:** Application Programming Interface
 - **Scraping:** técnica para extraer información de sitios web en forma automática y almacenarla en un formato estructurado.
- En la web, la transmisión de información se realiza mediante el protocolo HTTP (Hypertext Transfer Protocol)
 - HTTPS: versión más segura
 - **Modelo cliente-servidor:** un cliente establece una conexión, realizando una petición a un servidor y espera una respuesta del mismo.
- Para acceder a data en la web por estas 3 vías, utilizaremos algunas **librerías** específicas de Python (hay varias otras):
 - **requests:** <https://docs.python-requests.org/en/master/>
 - **BeautifulSoup:** <https://pypi.org/project/beautifulsoup4/>

MODELO DE COMUNICACIÓN CLIENTE-SERVIDOR

1. Enviar solicitud al servidor



2. Recibir respuesta del servidor

TRANSMISIÓN DE MENSAJES

Protocolo de transferencia de hipertexto **HTTP**



Una petición



Un número de
encabezados o **headers**



Una línea vacía



Un cuerpo de mensaje

Opcionalmente

EJEMPLO: LLAMADA A www.mipagina.cl

GET / HTTP/1.1

Una petición

Host: example.com

Connection: keep-alive

Cache-Control: max-age=0

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0

(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, Like Gecko)

Chrome/60.0.3112.90 Safari/537.36

Accept:

text/html, application/xhtml+xml, application/xml;q=0.9,*/*;q=0.8

Referer: https://www.mipagina.cl

Accept-Encoding: gzip, deflate

Accept-Language: en-US, en;q=0.8, nl;q=0.6

<CR><LF> Una línea vacía

Un número de
encabezados o headers

EJEMPLO: RESPUESTA DEL SERVIDOR WWW.MIPAGINA.CL

HTTP/1.1 200 OK

Una petición

Connection: keep-alive

Content-Encoding: gzip

Content-Type: text/html; charset=utf-8 Data: Mon, 28 Aug 2017

10:57:42 GMT Server: Apache v1.3 Vary:

Accept-Encoding

Un número de encabezados o headers

Transfer-Encoding: chunked

<CR><LF>

Una línea vacía

<html> <body>Bienvenidos a mi pagina web</body>

Un cuerpo de mensaje

</html>

Opcionalmente

MÉTODOS DE EXTRACCIÓN DE DATOS BASADOS EN HTTP



Descarga de datos desde una dirección URL
(Universal Resources Locator).



Extracción de datos desde APIs
(Application Programming Interface).



Scraping de páginas web

DATA EN LA WEB - URL

- Muchos datos de interés están publicados o disponibles en la Web.
 - Codificación de la descarga y extracción ➔ reproducibilidad, escalabilidad
- **URL:** apertura y descarga de datos a partir de Universal Resources Locator
 - url = protocolo (http, https) + nombre del recurso
 - Ej: Datos de Puntos BIP en datos.Gob.cl

```
'https://datos.gob.cl/dataset/c2969d8a-df82-4a6c-a1f8-e5eba36af6cf/resource/cbd329c6-9fe6-4dc1-91e3-a99689fd0254/download/pcma_20210901_oficio-4770_2013.xlsx'
```

- Método sencillo: usar funciones de pandas para leer archivos directamente desde la web
 - `pd.read_csv(url)`
 - `pd.read_excel(url)`

Ejemplo: Censo

Datos de proyección de la población de Chile hacia 2050,
<http://www.censo2017.cl>

URL

url='http://www.censo2017.cl/descargas/proyecciones/estimaciones-y-proyecciones-chile-1992-2050-base-2017-poblacion-e-indicadores.xlsx'



The screenshot shows a web browser window for the website censo2017.cl. The header features the 'CENSO 2017' logo and a search bar. On the right, there's a link to 'HISTORIA DEL CENSO'. The main content area has a red background with the text 'ENTREGA FINAL CENSO 2017' in white. Below this, a red banner reads 'Estimaciones y Proyecciones de la Población de Chile 1992-2050 (Total País)'. A descriptive paragraph explains the data provided by the National Institute of Statistics (INE). Under the heading 'Material Descargable', there are four download links: 'Estimaciones y proyecciones de la población de Chile 1992-2050 total país' (PDF), 'Síntesis de las estimaciones y proyecciones de la población de Chile 1992-2050 total país' (PDF), 'Estimaciones y proyecciones de la población de Chile 1992-2050 total país' (XLS), and 'Estimaciones y proyecciones de la población de Chile 1992-2050 total país' (CSV). Each download link is accompanied by a red 'Descarga' button.

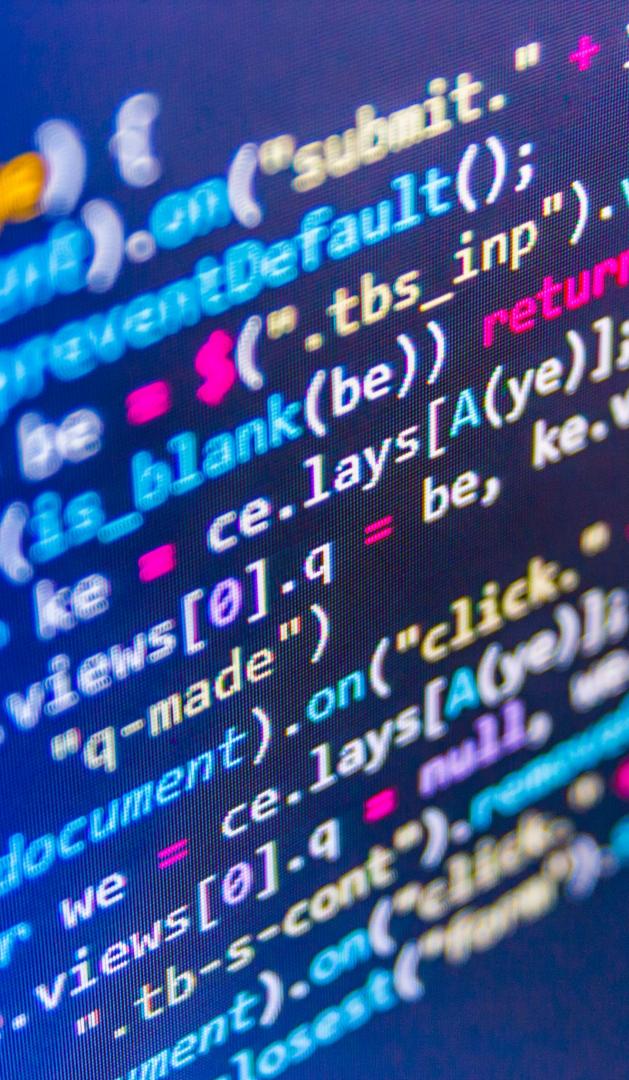
IMPLEMENTACIÓN DE DESCARGA DE DATOS EN pandas

Para conjuntos de datos estructurados:



Funciones de pandas

- `pd.read_csv(url)`
- `pd.read_excel(url)`



Librería requests



Librerías de Python proveen de funciones y clases para enviar peticiones HTTP (como `urllib`, `httpplib`, incluso `pandas`).



La librería **requests** implementa solicitudes HTTP para enviar peticiones a un servidor, mediante métodos: `requests.get()` y `requests.post()`



La respuesta del servidor es recogida en un objeto de la clase **requests.Response**, que implementa métodos y atributos para leer y explorar los datos extraídos.

IMPLEMENTACIÓN DE DESCARGA DE DATOS EN requests

```
import requests  
response=requests.get(url)
```

Solicitud tipo GET

```
output = open('proyeccionesCensoChile.xlsx', 'wb')  
output.write(response.content)  
output.close()
```

Escribir el contenido a un
archivo local

¿QUÉ ES UNA API?

Muchas páginas web o portales de datos cuentan con una **Interfaz de Programación de Aplicaciones** o **API** (Application Programming Interface) para facilitar el consumo de recursos disponibles por parte de sus usuarios.

API: Un conjunto de **protocolos y rutinas** para interactuar con aplicaciones de software

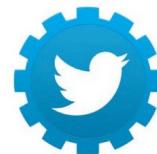
APIs are everywhere



Recibe solicitudes HTTP.



Entrega datos en un formato leíble por máquinas (JSON).





INTERACTUAR CON UNA API

Utilizamos la librería **requests** para enviar una solicitud y recibir un objeto de respuesta con datos solicitados



El método GET recibe la dirección del recurso.



También un conjunto de parámetros para personalizar la petición a la API.



A veces un conjunto de cabeceras o headers HTTP requeridos para acceder a la API.

Ejemplo: Datos Abiertos del Banco Mundial

Amplio catálogo de datos abiertos y varias API.

<http://data.worldbank.org>

THE WORLD BANK | Data

New to this site? [Start Here](#)

This page in: English Español Français العربية 中文

DataBank Microdata Data Catalog

World Bank Open Data

Free and open access to global development data

Search data e.g. GDP, population, Indonesia

Browse by [Country](#) or [Indicator](#)

MOST RECENT

Global metal markets: Weakening demand amid constrained supply? [»](#)
Wee Chian Koh, John Baffes, Jun 01, 2022

Conference round-up: the role of mobile data in global development research [»](#)
Maria Jones, Anya Marchenko, Jun 01, 2022

How to use PIP's Poverty Calculator page [»](#)
Martha Viveros, R. Andres Castaneda Aguilar, Tony Fujis, May 31, 2022

Food prices continued their two-year-[»](#)

[View all news](#) [View all blogs](#)

WHAT YOU CAN LEARN WITH OPEN DATA

Poverty headcount ratio at \$1.90 a day (2011 PPP) (% of population)

Extreme Poverty
The proportion of the world's population living in extreme poverty has dropped significantly

MULTILATERAL LEADERS TASK FORCE ON COVID-19 DATA DASHBOARD

IMF WORLD BANK GROUP World Health Organization WORLD TRADE ORGANIZATION

MLTF COVID-19 Data Dashboard [»](#)
Mar 23, 2022

Ejemplo: API Country, Banco Mundial

[API Country](#): entrega datos generales sobre todos los países del mundo, como su nombre, capital y nivel de ingresos

- Los parámetros y encabezados a configurar o definir son particulares a cada API, por lo que se recomienda siempre **revisar cuidadosamente la documentación de la API**, para así conocer todas las opciones de configuración de las peticiones.

```
import requests
url=' http://api.worldbank.org/v2/country'
response=requests.get(url),params={'format':'json'})
```

Ejemplo Banco Mundial: API Country

```
In[36]: response.json()
```

```
Out[36]: [ {'page' : 1, 'pages' : 6, 'per_page' : '50' , 'total' : 299},  
          [ { 'id' : 'ABW' ,  
              'iso2Code' : 'AW' ,  
              'name' : 'Aruba' ,  
              'region' : { 'id' : 'LCN' ,  
                          'iso2Code' : 'ZJ' ,  
                          'value' : 'Latin America & Caribbean ' },  
              'adminregion' : { 'id' : ' ' , 'iso2Code' : ' ' , 'value' : ' ' },  
              'incomeLevel' : { 'id' : ' HIC' , 'iso2Code' : 'XD' , 'value' : 'High income' },  
              'lendingType' : { 'id' : ' LNX' , 'iso2Code' : 'XX' , 'value' : 'Not classified' },  
              'capitalCity' : 'Oranjestad' ,  
              'longitude' : '-70.0167' ,  
              'latitude' : '12.5167' }]
```

Web Scraping



¿QUÉ ES WEB SCRAPING?

Automatización del proceso de búsqueda y extracción de datos de una página web



Puede afectar el tráfico y funcionamiento del servidor consultado, no todos los sitios lo permiten.



Cada sitio tiene asociado un archivo de texto (robots.txt) que indica si está permitido o qué partes del sitio se puede rastrear.

Ejemplo: robots.txt de Wikipedia

```
# robots.txt for http://www.wikipedia.org/ and friends
#
# Please note: There are a lot of pages on this site, and there are
# some misbehaved spiders out there that go _way_ too fast. If you're
# irresponsible, your access to the site may be blocked.
#
# Observed spamming large amounts of https://en.wikipedia.org/?curid=NNNNNNN
# and ignoring 429 ratelimit responses, claims to respect robots:
# http://mj12bot.com/
User-agent: MJ12bot
Disallow: /

# advertising-related bots:
User-agent: Mediapartners-Google*
Disallow: /

# Wikipedia work bots:
User-agent: IsraBot
Disallow:
User-agent: Orthogaffe
Disallow:

# Crawlers that are kind enough to obey, but which we'd rather not have
# unless they're feeding search engines.
User-agent: UbiCrawler
Disallow: /
```

PRINCIPIOS ÉTICOS PARA WEBSRAPING



**Preferir uso de
APIs**

Si es que
existen



**Enviar peticiones
a tasas
razonables**

Sólo extraer
datos realmente
necesarios



**Retribuir el sitio,
en lo posible**

Por ejemplo,
citando en
publicaciones

WEBSRAPING: ETAPAS DEL PROCESO

¿Cómo realizamos el web scraping?



Identificar el sitio web sobre el cual se quiere hacer scraping

Analizar su código fuente para ubicar los datos a extraer.

Descargar todo el código fuente

De una determinada página del sitio.

Programar rutina para extraer datos de interés del código fuente

Almacenar como datos estructurados.

Primera etapa:

Identificar la página web y analizar su código fuente.

HTML es un formato semi estructurado, se caracteriza por uso de etiquetas o tags anidadas que definen las secciones y elementos de la página web, y sus atributos o características de visualización.

CÓDIGO HTML



**Comienza y
termina con las
etiquetas <html>**



**Suele contener
un encabezado**

Identificado por
la etiqueta
<head>



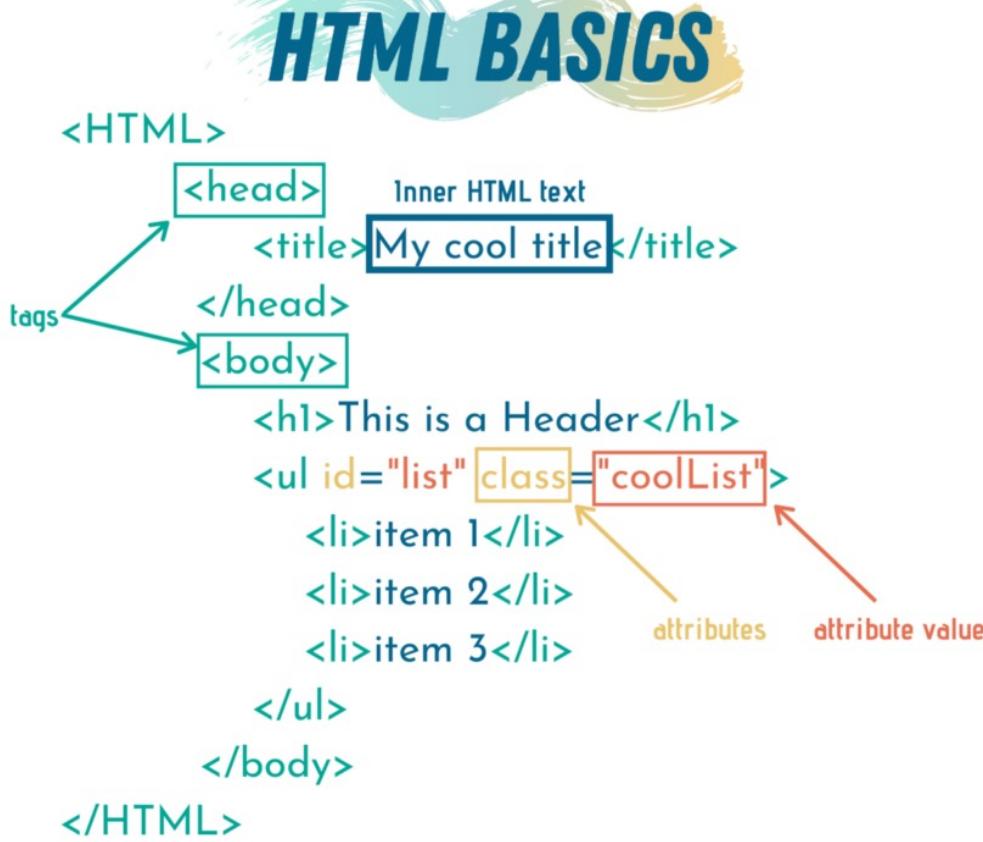
**Contiene un
cuerpo o <body>**



**Contiene otros
elementos**

Listas (),
títulos (<title>),
párrafos (<p>),
etc.

CÓDIGO HTML



TOP 10 HTML TAGS

TAG	USAGE
<div>	used to create a division in the page
	used to add a span around some items
 / 	ol >> ordered list ul >> unordered list li >> list item
<table> <th> <tr>	table >> creates a table th >> table header tr >> table row
<div>	used to create a division in the page
	used to add a span around some items
 / 	ol >> ordered list ul >> unordered list li >> list item
<table> <th> <tr>	table >> creates a table th >> table header tr >> table row

**Los navegadores
Google Chrome, Firefox
o Explorer cuentan con
herramientas de
inspección y desarrollo
HTML.**



FORMA DE ACCEDER AL CÓDIGO FUENTE

The screenshot shows a web browser window for the Pontificia Universidad Católica de Chile (uc.cl) website. The page content is about a project called "PROYECTO INES CIENCIA ABIERTA" where UC is looking to make research results open access. A context menu is open over the text, with the option "Ver el código fuente de la página" highlighted in blue.

Ir al inicio | ADMISIÓN | Covid-19 | Medios | Biblioteca | Donaciones | Mi Portal UC | Correo | English site

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

Programas de estudio | Investigación | Facultades | Internacionalización | Extensión | Universidad | Información para | Q

Investigación

PROYECTO INES CIENCIA ABIERTA:

UC busca poner a libre disposición los resultados de investigación

Ir al artículo →

Atrás
Reenviar
Volver a cargar
Guardar como...
Imprimir...
Transmitir...
Buscar imágenes con Google Lens
Crear un código QR para esta página
Traducir a español
Ver el código fuente de la página
Inspeccionar

03 04 II

La forma más sencilla de acceder al código fuente de una página es hacer clic derecho y seleccionar la opción llamada “Ver código fuente” o similar.

ANALICEMOS: PÁGINA WEB DE EJEMPLO

```
<html>
  <head>
    <title>IMT1001 Taller 4: Web Scrapping</title>
  </head>
  <body>
    <h1>1. Código HTML</h1>
    <p> Este es un ejemplo para comprender la estructura de un documento HTML. Los pasos a seguir
       para extraer los datos de interés son:
      <ul id='list' class='pasos'>
        <li> Leer el código HTML</li>
        <li> Formatear (parse) usando BeautifulSoup, y </li>
        <li> Extraer la información de interés. </li>
      </ul>
    </p>

    <p>Por ejemplo, podemos extraer párrafos de texto, los cuales pueden ser extensos, o muy
       breves. Como comentamos en clase, los cuerpos de texto que se extraen de la web pueden ser
       utilizados por ejemplo para construir grandes datasets de entrenamiento de modelos de
       lenguaje. Esto es una excelente motivación para aprender a hacer webscraping!!</p>

    <p> </p>
    <p> También podemos crear una tabla:
      <p> </p>

      <table>
        <tr>
          <th>Curso</th>
          <th>Créditos</th>
          <th>N Estudiantes</th>
        </tr>
        <tr>
          <td>IMT1101</td>
          <td>10</td>
          <td>42</td>
        </tr>
        <tr>
          <td>OFG1100</td>
          <td>5</td>
          <td>25</td>
        </tr>
      </table>
    </p>
  </body>
</html>
```

DESCARGAR EL CÓDIGO FUENTE DE UNA PÁGINA WEB

- Ahora que ya conocemos la página web y ubicamos los datos a extraer, necesitamos descargar todo el código fuente .
- Esto consiste en enviar una solicitud al servidor, para que nos envíe de vuelta un texto HTML que contiene toda la estructura y formato de la página.

```
import requests  
  
url='https://www.mipaginadedatos.com'  
  
resp=requests.get(url)  
  
html_code=resp.text
```

EXTRAER DATOS DE INTERÉS DEL CÓDIGO FUENTE

- El código fuente de una página puede ser bastante complejo, y contener muchas secciones y elementos distintos. Podemos pensar que es una especie de “sopa” de etiquetas o tags.



Fuente: Analyticsvidhya. (2021) [A simple introduction to web scraping with beautiful soup.](#)

**En Python existen librerías que
permiten extraer elementos útiles
de esta sopa HTML.**

LIBRERÍA BeautifulSoup

```
from bs4 import BeautifulSoup  
  
soup = BeautifulSoup(html_code)
```

```
In [7]: 1 soup.head
```

```
Out[7]: <head>  
        <title>Clase 4: Web Scrapping</title>  
        </head>
```

```
In [10]: 1 soup.title
```

```
Out[10]: <title>Clase 4: Web Scrapping</title>
```

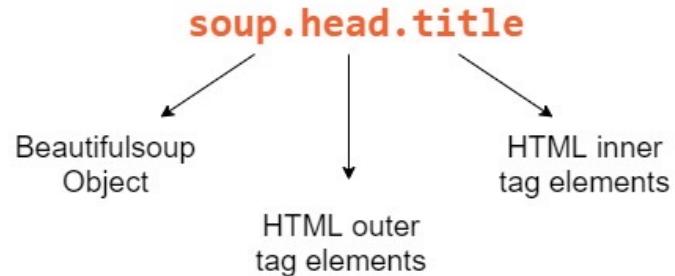


TABLA DE DATOS

```
In [11]: 1 | soup.table
```

```
Out[11]: <table>
<tr>
<th>Curso</th>
<th>Créditos</th>
<th>N Estudiantes</th>
</tr>
<tr>
<td style="text-align: center; vertical-align: middle;">IMT21001042IMT22001025OFG1100535OFG1200528
```

EXTRAER Y ALMACENAR DATOS EN UN DATAFRAME

```
In [11]: 1 import pandas as pd
2 df=pd.DataFrame(columns=['Curso','Creditos','nEstudiantes'])
3
4 table=soup.find('table')
5
6 rows=table.find_all('tr')
7
8 for row in rows[1::]:
9     cols = row.find_all("td")
10    col_text=[c.text for c in cols]
11    df=df.append({'Curso':col_text[0],'Creditos':col_text[1],'nEstudiantes':col_text[2]},ignore_index=True)
12 df
```

	Curso	Creditos	nEstudiantes
0	IMT2100	10	42
1	IMT2200	10	25
2	OFG1100	5	35
3	OFG1200	5	28