IMT2200

INTRODUCCIÓN A CIENCIA DE DATOS

2022-2

FUENTES DE DATOS

CLASE 5

AGENDA

- Repaso Clase 4
- Fuentes de Datos
- Extracción de datos desde la Web

Datos Semi-Estructurados

DATOS SEMI-ESTRUCTURADOS

XML: Extensible Markup Language

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

```xml
<?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`

```json
{
  "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"
    }
  ]
}
```

Notebooks Clase 4, Actividad 1

Fuentes de Datos

CLASIFICACIONES DE LAS FUENTES DE DATOS

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

DATOS ABIERTOS

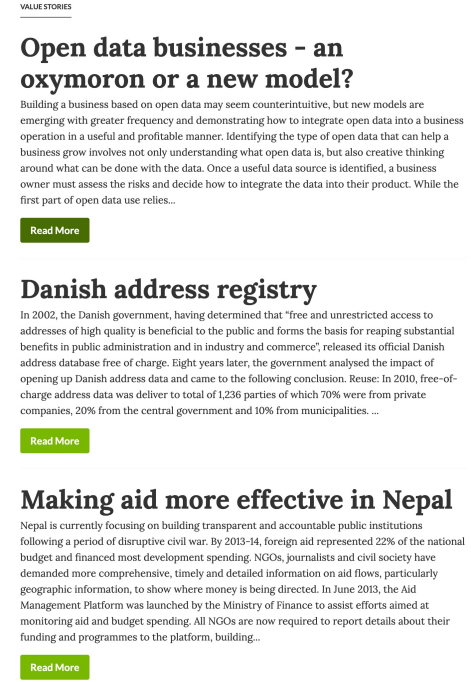
- Datos disponibles en **forma gratuita y sin restricciones** de derechos de autor (copyright), patentes u otros mecanismos de control
- Pueden ser **utilizados, reutilizados y redistribuidos** libremente por cualquier persona
- Sujetos, cuando más, al requerimiento **de atribución** y de **compartirse** 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.

Open Knowledge Foundation
https://okfn.org/opendata/

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 deliver 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

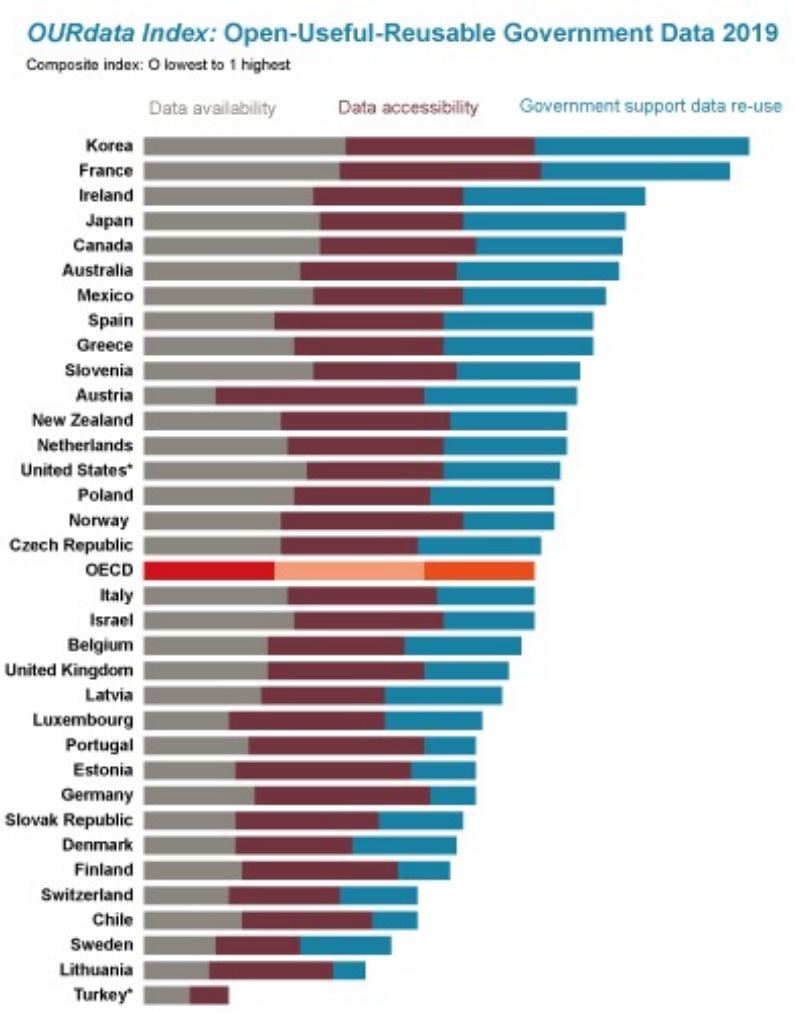
http://opendatahandbook.org/value-stories/en/

DATOS ABIERTOS

Datos Abiertos de Gobierno

Filosofía y conjunto de políticas que promueven la transparencia, responsabilidad y creación de valor económico y social mediante la disponibilización de datos de gobierno.

https://www.oecd.org/gov/digital-government/open-government-data.htm



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 propietarios

DATA EN 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/

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'
```

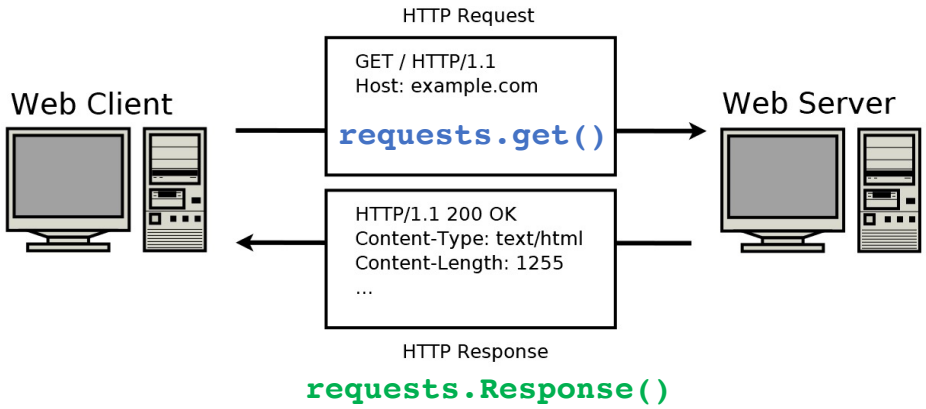
  - Método sencillo: usar funciones de pandas para leer archivos directamente desde la web
    - `pd.read_csv(url)`
    - `pd.read_excel(url)`

`requests`

- HTTP define un conjunto de métodos de petición:
  - **GET:** el método GET solicita datos a un recurso específico.
  - POST: envía data al servidor para crear o actualizar un recurso.

- `Requests:`
  - Implementa solicitudes HTTP (GET, POST) para enviar peticiones a un servidor
    - `requests.get()`
    - `requests.post()`
  - Recoge la respuesta en un objeto tipo `'Response',` que implementa métodos y atributos para leer y explorar los datos extraídos.
    - `response = requests.get(url)`
    - `reponse`

1. Enviar solicitud al servidor



2. Recoger respuesta y leer los datos extraídos

DATA EN LA WEB - API

- **API:** Application Programming Interface
  - Conjunto de protocolos y rutinas para interactuar con aplicaciones de software (o entre softwares)
  - Reciben solicitudes y entregan datos en formato leíble por máquinas (JSON).

```
import requests
url = 'http://www.omdbapi.com/?t=hackers'
r = requests.get(url)
json_data = r.json()
for key, value in json_data.items():
    print(key + ':', value)
```

- http - making an HTTP request
- **www.omdbapi.com** - querying the OMDB API
- `?t=hackers`
  - Query string
  - Return data for a movie with title (t) 'Hackers'

`'http://www.omdbapi.com/?t=hackers'`

APIs are everywhere

UBER Developers

Instagram API