



Almacenamiento y captura de datos

Claudio Aracena

GobLab - Universidad Adolfo Ibáñez
Chatbot Chile



Presentación

- Ingeniero Civil Industrial, U. de Chile
- Magíster en Tecnologías de la Información, U. de Sydney
- He trabajado en WIC (UCHile), Data61 (CSIRO, Australia) y CMM (UCHile)
- Científico de datos, GobLab UAI
- Co-fundador Chatbot Chile

Contacto

claudio.aracena@uai.cl

<https://www.linkedin.com/in/caracena2/>

<https://github.com/caracena>



Introducción al curso

- Clases expositivas y prácticas (Python y SQL)
- 1 evaluación (tarea)
- Para aprobar:
Nota mínima de 4.0 y 60% asistencia



Contenidos

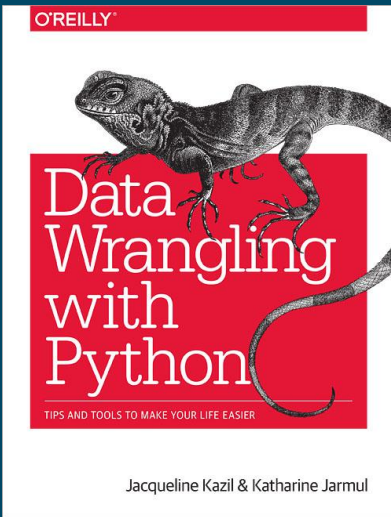
- Captura de datos desde archivos
- Base de datos
- Captura y almacenamiento de datos en BD
- Captura de datos de la Web (Web scraping)
- Captura de datos de API (ej: Twitter)
- Captura y almacenamiento en arquitecturas Big data





Recursos

- Data Wrangling with Python. Tips and Tools to Make Your Life Easier. By Jacqueline Kazil, Katharine Jarmul (2016).



Códigos y clase en:

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

Clase de hoy



Captura de datos desde archivos

- Archivos de texto
- CSV
- JSON
- Excel





Archivos de texto

- Lectura

with open('file.txt') as f:

```
lines = f.readlines()
```

```
content = f.read()
```

- Escritura

with open('file.txt', 'w') as fw:

```
fw.write('texto')
```



Ejercicio

Lea el archivo de texto (names.txt) y cree un archivo csv (names.csv) con 1 columna de nombres y otra de apellidos.

funciones útiles:

- `str.split()` : separa un string str
- `str.join()`: une una lista usando str



Ejercicio

Cuente la frecuencia de las palabras del archivo de texto `el_quijote.txt` y muestre las 10 palabras más frecuentes.

funciones útiles:

- 1) `sorted()` para ordenar un diccionario
- 2) usar `collections.Counter`



CSV

- Lectura

with open('file.csv') as f:

```
    reader = csv.reader(f)
```

```
    for row in reader:
```

```
        print(row)
```

- Escritura

with open('file.csv', 'w') as fw:

```
    writer = csv.writer(fw)
```

```
    writer.writerow(row)
```



CSV en pandas

- Lectura

```
import pandas as pd
```

```
df = pd.read_csv('file.csv')
```

- Escritura

```
df.to_csv('file.csv')
```



JSON

- Lectura

```
import json
```

```
with open('file.json') as f:
```

```
    content = f.read()
```

```
    data = json.loads(content)
```

```
for item in data:
```

```
    print(item)
```

- Escritura

```
import json
```

```
with open('file.json', 'w') as f:
```

```
    json.dump(data, f)
```



JSON en pandas

- Lectura

```
import pandas as pd
```

```
df = pd.read_json('file.json')
```

- Escritura

```
df.to_json('file.json')
```



Excel

- Lectura

```
import xlrd
```

```
book = xlrd.open_workbook("file.xlsx")
```

```
sheet = book.sheet_by_index(0)
```

```
sheet = book.sheet_by_name("Hoja 1")
```

```
row = sheet.row_values(0)
```



Excel en pandas

- Lectura

```
import pandas as pd
```

```
pd.read_excel("file.xlsx", sheet_name=0)
```



Ejercicio

Cree un dataframe a partir de los datos del archivo SOWC 2014 Stat Tables_Table 9.xlsx. Las columnas del dataframe deben ser:

```
columns = ['Name', 'child_labor_total', 'child_labor_male', 'child_labor_female',  
           'child_marriage_married_by_15', 'child_marriage_married_by_18']
```