

TECNICATURA SUPERIOR EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

PROYECTO INTEGRADOR

Cotizacion Dolar oficial

Autores

Beverly Gonzalez Casanova

Sebastian Galvan

Elin Yanel Gisinger

Año 2022

Índice

1. Objetivos	2
2. Fundamentación	2
3. Desarrollo del proyecto	2
3.1. Python	3
3.2. Jupyter Notebooks	4
3.3. SQL	6
3.4. Numpy y pandas	7
3.5. Power Bi	8
3.6. Gestión del proyecto	9
3.7. Enlaces del Proyecto	9

1. Objetivos

En este proyecto tecnológico integrador se propone vivenciar una experiencia de trabajo colaborativo real en Ciencia de Datos a través de un entorno ágil, mediante trabajo en equipo, y aplicando los conocimientos adquiridos a través de los espacios curriculares cursados en el año, sumado a la investigación exhaustiva de los saberes para afrontar este proyecto para optimizar la creatividad y calidad del proyecto. .

2. Fundamentación

La Ciencia de datos consiste en el estudio de los datos con el fin de extraer información significativa para empresas obtenidos de múltiples fuentes. Es un enfoque multidisciplinario que combina principios y prácticas del campo de la estadística, las matemáticas y la inteligencia artificial para analizar grandes cantidades de datos. Este análisis permite a los científicos de datos plantear y responder a preguntas como “qué pasó”, “por qué pasó”, “qué pasará” y “qué se puede hacer con los resultados”. Una de las técnicas suele ser el “Scraping”, que es una forma de minería de datos no estructurada, que cual permite obtener datos de páginas web, escaneando su código HTML y generar patrones de extracción de datos, para luego manipular esta información desde la aplicación desarrollada.

3. Desarrollo del proyecto

El primer paso para extraer información es identificar correctamente el elemento o elementos que contienen la información deseada, por lo que en las páginas de EL cronista [1] y dolarHoy [2] trabajamos inspeccionando el elemento, quedándonos con la información disponible (etiquetas, o los atributos class y/o id) para luego ser utilizadas en BeautifulSoup.

Con Python podemos crear programas que generen peticiones al servidor y recibir el código fuente en formato HTML.

Antes de la toma de datos con web Scraping, se desarrolló una aplicación en Python, en donde se utilizaron librerías tales como BeautifulSoup (la última versión) que nos permitió extraer información de contenido en formato HTML. Para usarla, fue necesario especificar un parser, quien fue el responsable de transformar el documento HTML a un árbol complejo de objetos Python. Para descargar el contenido de la página se utilizó la librería requests. Uno de los parsers utilizados para extraer información es el lxml.

Se utilizó GitHub como plataforma de desarrollo colaborativo, para alojar los proyectos utilizando el sistema de control de versiones Git, en donde se muestran los datos almacenados y en donde se trabajó la mayor parte del proyecto.

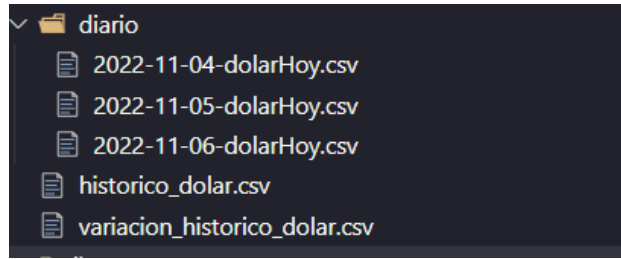


Figura 1: mes

3.1. Python

Utilizamos Python como lenguaje para realizar el web scraping, lo primero que hicimos fue definir un entorno virtual para poder instalar las librerías que utilizamos en el proyecto, utilizamos Virtualenv para ello, tambien utilizamos Pip que es una herramienta que nos facilita la tarea de instalar y manejar dependencias en el proyecto, creamos un archivo requirements.txt para dejar constancia de las librerías utilizadas y sus respectivas versiones. Para el scraping utilizamos varias estructuras que aprendimos durante la capacitación en Cisco, por ejemplo, variables, arrays y diccionarios. Usamos estructuras condicionales y de iteración para trabajar con los datos obtenidos de las paginas y definimos módulos para poder importar archivos y funciones. Entre las librerías externas a la librería estándar de Python que utilizamos podemos mencionar a request, beautifulsoup, sqlite3, lxml y pandas. Tambien utilizamos datetime para trabajar con fechas.

```
1 for dolar in dolares:
2     # Me quedo con los divs con la clase tile is-child
3     titles = dolar.find_all('div', class_="tile is-child")
4     for title in titles:
5         # Me quedo con los los elementos de tipo a "links"
6         a = title.find_all('a')
7         for dolar in a:
8             if dolar != []:
9                 # Dentro de cada elemento a ahy elementos divs con los datos de cada tipo de dolar
10                # El nombre del tipo de dolar, el valor para la compra y el de vetnta
11                p1 = dolar.find('div', class_="title")
12                p2 = dolar.find('div', class_="compra")
13                p3 = dolar.find('div', class_="venta")
14
15                if p1 != " ":
16                    #Los agrego al diccionario
17                    dolarDic["tipo"].append(p1.get_text())
18                    dolarDic["compra"].append(p2.get_text())
19                    dolarDic["venta"].append(p3.get_text())
20
21 fecha = datetime.now().date()
22 dml_dolar_hoy.insertarDiarios(dolarDic)
23 df = pd.DataFrame(dolarDic)
24 df.to_csv(f'../data/diario/{fecha}-dolarHoy.csv', index=False, encoding='utf-8')
25 return df
```

Figura 2: Código Python

3.2. Jupyter Notebooks

Decidimos usar notebooks de Jupyter para juntar las funcionalidades de scripting, la creación de archivos csv, así como la creación de bases de datos y el alta y borrado de registros en la misa.

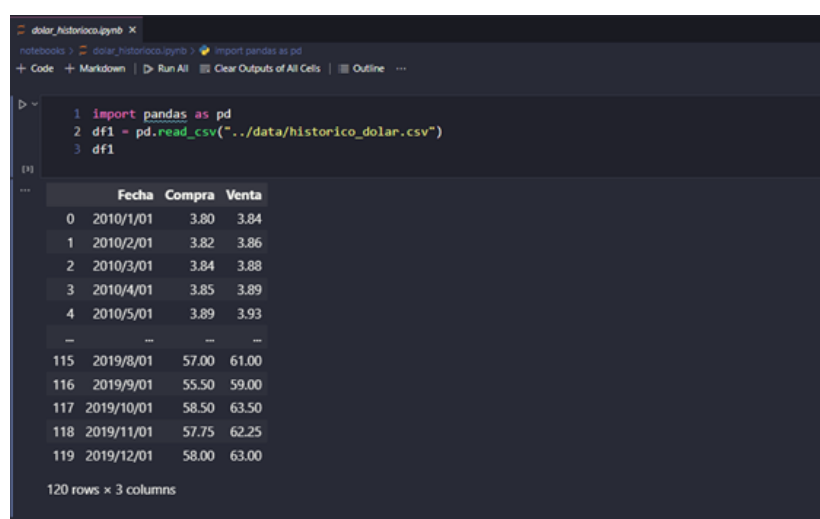
¿Qué son las Jupyter notebooks? Las Notebook de Jupyter nos permites crear y compartir documentos que contengan código activo, ecuaciones, visualizaciones y texto.

Jupyter Notebooks es un proyecto derivado del proyecto IPython, que solía tener un proyecto IPython Notebook. El nombre, Jupyter, proviene de los principales lenguajes de programación compatibles que admite: Julia, Python y R. Jupyter se instala con el kernel IPython, que permite escribir programas en Python, pero actualmente hay más de 100 otros kernels que se pueden usar.

En nuestro caso utilizamos ipykernel en su versión 6.17.0, se encuentra especificado dentro del archivo requirements.txt, así como las demás librerías que utiliza Jupyter y el proyecto.

En el proyecto definimos 3 notebooks:

- general.ipynb: esta contiene celdas de ejecución para realizar el scraping diario del dólar, el cual obtiene 5 cotizaciones, crea el archivo csv correspondiente, y las guarda en una base de datos SQLite registrando, fecha actual, tipo de cotización, el valor de compra y el valor de venta.
- notebooks/doalrHoy.ipynb: este me permite abrir un archivo csv del scraping del día y obtener sus datos dentro de un Data Frame de la librería Pandas.
- notebooks/dolar_historico.ipynb: esta notebook me permite abrir el archivo csv resultado de hacer el scraping de datos históricos del dólar en las 2 ultimas décadas agregarlos a un Data Frame y visualizar los datos, en esta notebook hay otra celda para visualizar con el mismo procedimiento anterior las variaciones trimestrales, semestrales y anuales del dólar.



```
1 import pandas as pd
2 df1 = pd.read_csv("../data/historico_dolar.csv")
3 df1
```

	Fecha	Compra	Venta
0	2010/1/01	3.80	3.84
1	2010/2/01	3.82	3.86
2	2010/3/01	3.84	3.88
3	2010/4/01	3.85	3.89
4	2010/5/01	3.89	3.93
...
115	2019/8/01	57.00	61.00
116	2019/9/01	55.50	59.00
117	2019/10/01	58.50	63.50
118	2019/11/01	57.75	62.25
119	2019/12/01	58.00	63.00

120 rows x 3 columns

Figura 3: Registro histórico desde una Notebook de Jupyter

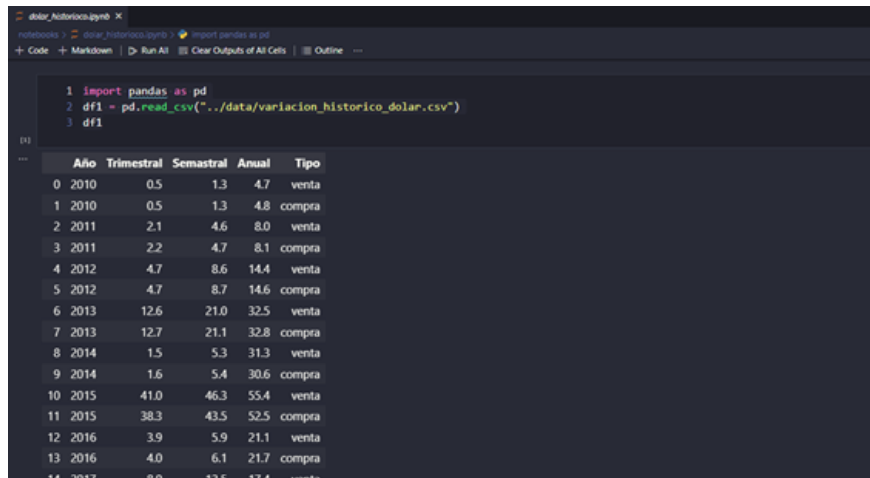


Figura 4: Variaciones histórico dolar oficial.

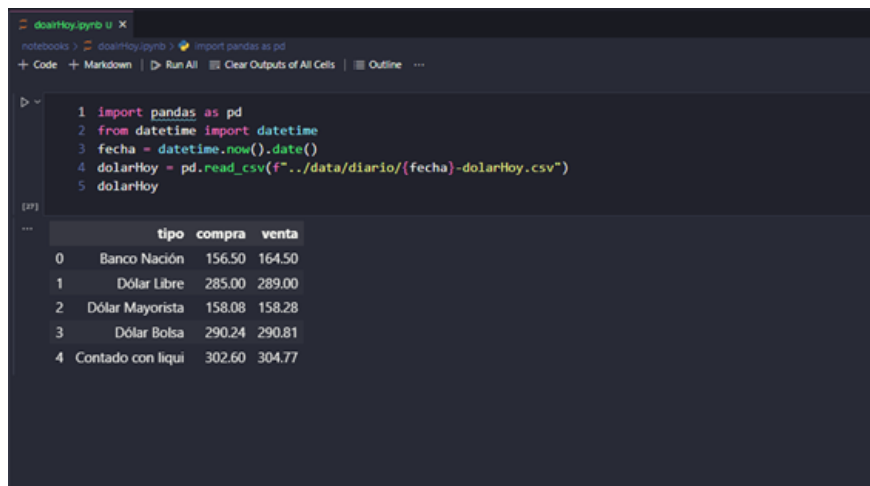


Figura 5: Velicación de cotización del dolar en diferentes entidades

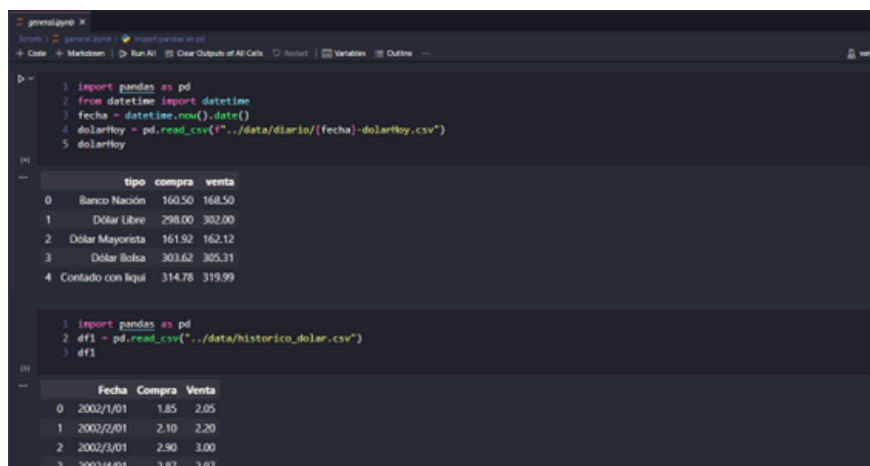


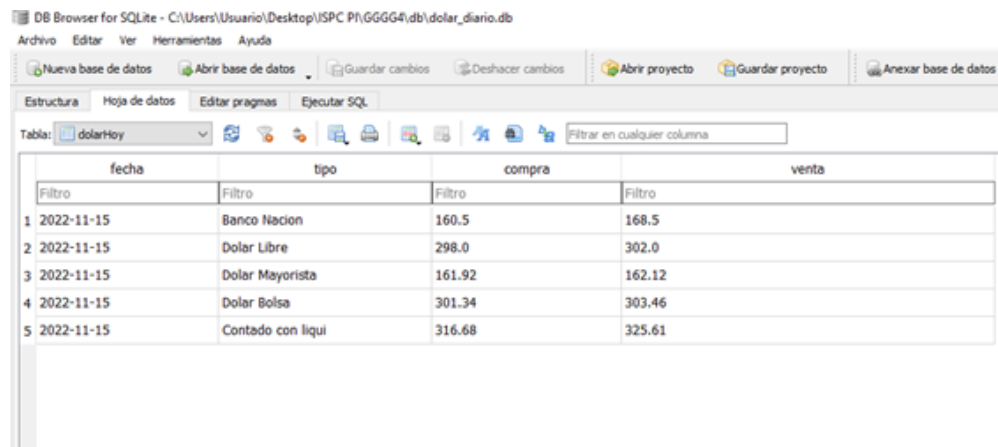
Figura 6: Nootbook de operaciones generales

3.3. SQL

Para el apartado de SQL decidimos utilizar SQLite, este uno de los sistemas de bases de datos relacionales más populares y fáciles de usar. Posee muchas características sobre otras bases de datos relacionales. SQLite es un sistema de gestión de bases de datos relacionales integrado y sin servidor. Es muy conveniente ya que tiene menos de 500kb de tamaño, que es significativamente menor que otros sistemas de administración de bases de datos como MySQL o PostgreSQL.

En nuestro proyecto utilizamos 2 bases de datos:

- dolar_historico.db: en esta base de datos guardamos los registros históricos de los valores de compra y venta del dólar oficial, la misma se crea al hacer es scraping de la pagina a la cual consultamos, de este proceso obtenemos 2 archivos CSV, uno con los valores antes descriptos y otro que almacena las variaciones trimestrales, semestrales y anuales promedios de los precios del dólar, diferenciados por el tipo, compra o venta, y el año que se está calculando la variación.
- dolar_diario.db: esta base de datos registra en una tabla la cotización del dólar diariamente sobre cinco entidades diferentes (Banco Nación, dólar Libre, dólar Mayorista, dólar Bolsa y Contado con liqui), valor de compra y venta y fecha en la que fue consultado.



	fecha	tipo	compra	venta
1	2022-11-15	Banco Nacion	160.5	168.5
2	2022-11-15	Dolar Libre	298.0	302.0
3	2022-11-15	Dolar Mayorista	161.92	162.12
4	2022-11-15	Dolar Bolsa	301.34	303.46
5	2022-11-15	Contado con liqui	316.68	325.61

Figura 7: Base de datos registros diarios

DB Browser for SQLite - C:\Users\Usuario\Desktop\USPC P\GGGG4\db\dolar_historico.db

Archivo Editar Ver Herramientas Ayuda

Nueva base de datos Abrir base de datos Guardar cambios Deshacer cambios Abrir proyecto Guardar proyecto Anexar base de datos

Estructura Hoja de datos Editar pragmas Ejecutar SQL

Tabla: historico

	Año	Mes	Compra	Venta
1	2010	enero	3,80	3,84
2	2010	febrero	3,82	3,86
3	2010	marzo	3,84	3,88
4	2010	abril	3,85	3,89
5	2010	mayo	3,89	3,93
6	2010	junio	3,89	3,93
7	2010	julio	3,90	3,94
8	2010	agosto	3,91	3,95
9	2010	saptiembre	3,92	3,96
10	2010	octubre	3,92	3,96
11	2010	noviembre	3,95	3,99
12	2010	diciembre	3,94	3,98
13	2011	enero	3,97	4,01
14	2011	febrero	3,99	4,03
15	2011	marzo	4,01	4,05
16	2011	abril	4,04	4,08
17	2011	mayo	4,05	4,09

Figura 8: Datos Históricos dolar oficial

DB Browser for SQLite - C:\Users\Usuario\Desktop\USPC P\GGGG4\db\dolar_historico.db

Archivo Editar Ver Herramientas Ayuda

Nueva base de datos Abrir base de datos Guardar cambios Deshacer cambios Abrir proyecto Guardar proyecto Anexar base de datos

Estructura Hoja de datos Editar pragmas Ejecutar SQL

Tabla: variaciones

	Año	Trimestral	Semestral	Anual	Tipo
1	2010	0,5	1,3	4,8	compra
2	2011	2,2	4,7	8,1	compra
3	2012	4,7	8,7	14,6	compra
4	2013	12,7	21,1	32,8	compra
5	2014	1,6	5,4	30,6	compra
6	2015	38,3	43,5	52,5	compra
7	2016	4,0	6,1	21,7	compra
8	2017	7,6	12,2	17,2	compra
9	2018	8,7	29,6	100,0	compra
10	2019	4,5	39,8	57,6	compra
11	2010	0,5	1,3	4,7	venta
12	2011	2,1	4,6	8,0	venta
13	2012	4,7	8,6	14,4	venta
14	2013	12,6	21,0	32,5	venta
15	2014	1,5	5,3	31,3	venta
16	2015	41,0	46,3	55,4	venta

Figura 9: Variaciones dolar histórico.

3.4. Numpy y pandas

En el proyecto utilizamos la librería Pandas, que trabaja con la librería Numpy internamente. Pandas se define como una biblioteca de código abierto que proporciona manipulación

de datos de alto rendimiento en Python. Como mencionamos está construido sobre el paquete Numpy, lo que significa que se requiere Numpy para operar con Pandas. El módulo Pandas trabaja principalmente con datos tabulares, es decir datos ordenados en forma de tabla, muy similar a una hoja de cálculo, este proporciona algunos conjuntos de herramientas poderosas como dataframes y series, que se utilizan principalmente para análisis de datos. Una vez obtenidos los datos del scraping y creado el respectivo diccionario, utilizamos pandas para construir los dataframes, estos son una estructura de datos ampliamente utilizada que funcionan como una matriz bidimensional con ejes etiquetados (filas y columnas), para posteriormente utilizar las funciones de Pandas para crear los archivos CSV, y luego las tablas en la base de datos.

Se crearon gráficos para los valores de dólar Histórico, así como también los datos para la compra y venta de los días 04/11 a 15/11 del 2022.

A screenshot of a code editor window with a dark background and light-colored text. The code is written in Python and uses the Pandas library. It shows two dataframes being created from dictionaries, then each being saved to a CSV file using the 'to_csv' method. After that, the data is loaded back into dataframes using 'read_csv' and then inserted into a database using 'to_sql'. The database connection is established using 'con'.

```
1 df = pd.DataFrame(hist_diccionario)
2 df.to_csv(f'../data/historico_dolar.csv', index=False, encoding='utf-8', header=labels_tabla_historioco)
3
4 df_var = pd.DataFrame(variaciones_diccionario)
5 df_var.to_csv(f'../data/variacion_historico_dolar.csv', index=False, encoding='utf-8', header=labels_tabla_variacion)
6
7 con.execute("DELETE FROM historico")
8 df = pd.read_csv("../data/historico_dolar.csv")
9 df.to_sql("historico", con, if_exists='append', index=False)
10
11
12 con.execute("DELETE FROM variaciones")
13 df = pd.read_csv("../data/variacion_historico_dolar.csv")
14 df.to_sql("variaciones", con, if_exists='append', index=False)
```

Figura 10: Libreria Pandas. Funciones to_csv y to_sql

3.5. Power Bi

Como otras tecnologías optamos por trabajar con Power Bi, que es un servicio de análisis de datos de Microsoft orientado a proporcionar visualizaciones interactivas y capacidades de inteligencia empresarial.

Tomando los archivos CSV, se procedió a analizar mediante gráficos la compra y venta del histórico del Dólar de los meses y años desde el 2002 a 2019, y además, los valores de dólar Histórico, así como también los datos para la compra y venta de los días 04 a 15 del presente mes.

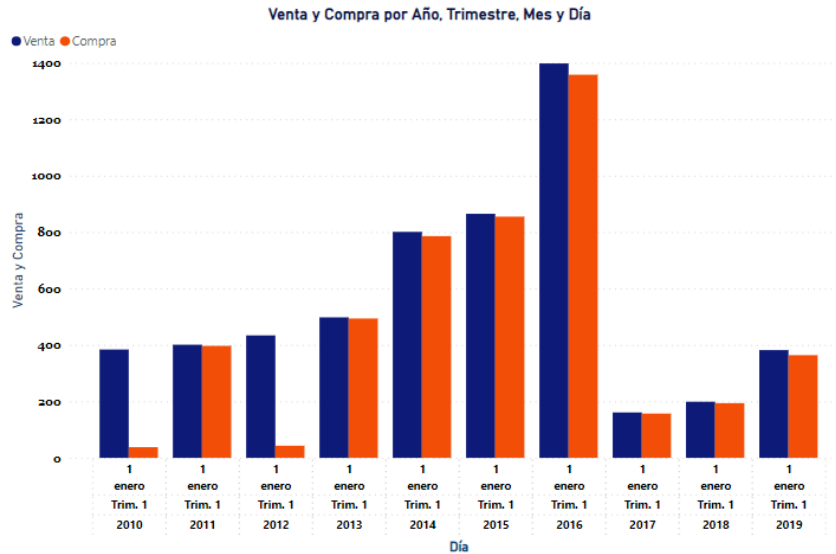


Figura 11: Gráfico dolar histórico. Se tomaron los datos del 1 de enero de cada año desde 2010 a 2019.

3.6. Gestión del proyecto

Se creó un repositorio en GitHub con el nombre del grupo GGGG4 [3], donde subimos los documentos del proyecto, los códigos y las bases utilizadas.

Por otro lado, se creó también un tablero de Trello [4] donde se organizó, planificó y se compartieron enlaces y recursos que se requerían para cada una de las actividades. Se compartió el acceso a ispc.programador2022@gmail.com.

3.7. Enlaces del Proyecto

[1] Enlace Biblioteca digital profesional - SIL (Editorial Sistema de Informes Laborales, Impuestos y Sociedades):

<http://www.sil1.com.ar>

<http://www.sil1.com.ar/soft/LABORAL/CARPETA4/in7-2010.htm>

[2] Enlace DolarHoy: <https://dolarhoy.com>

[3] Enlace GitHub: <https://github.com/ispc-programador2022/GGGG4.git>

[4] Enlace Trello: <https://trello.com/b/BojVNVdY/proyecto-integrador-ispc>