

DASH

Dra. Yessenia Yari Ramos



Universidad Católica
San Pablo

Departamento de Ciencia
de la Computación

Contenido

- Qué es Dash
- Enfoque Principal de Dash
- Características de Dash
- Conexión a la base de datos
- Pasos importantes para trabajar con Dash
- Construcción de un aplicativo con Dash
- Formas de diseñar

¿Qué es Dash?

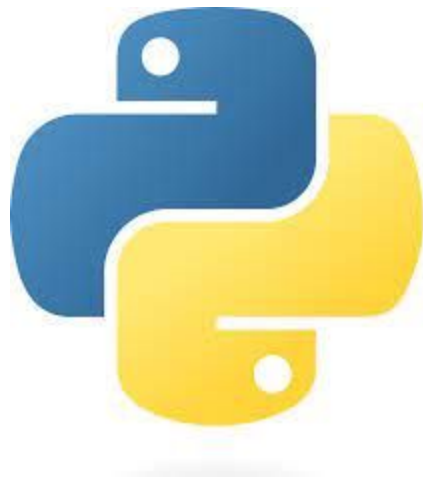
Dash

Es un framework de código abierto desarrollado por **Plotly**

Permite **crear aplicaciones webs** interactivas de manera rápida y sencilla

Utiliza lenguajes de programación como Python

Dash está basado principalmente en Flask, Plotly, ReactJS



Enfoque principal de Dash



Creación de aplicaciones de análisis de datos y paneles de control (dashboards), que permite a los usuarios visualizar y explorar datos de manera interactiva



Características de Dash



Características de Dash

1. Aplicación renderizada en navegador

Aplicación desplegado en un servidor

Compartir nuestro dashboard a traves del URL

2. Python-Centric

Lenguaje principal python

Integración con bibliotecas de análisis de datos (Plotly, Panda, Numpy)

3. Componentes interactivos

Gráficos, tablas, controles deslizantes y más

Podemos personalizar el estilo de nuestras aplicaciones, ya que usa CSS

Características de Dash

4. Aplicación Multiusuario

Múltiples usuarios con sesiones individuales (interacción simultánea)

5. Estructura Declarativa

Fácil de crear aplicaciones, puedes definir el interfaz de usuario y la lógica de aplicación

6. Visualización de datos de alta calidad

Gráficos

Mapas de color

7. Opensource

Pasos importantes para trabajar con Dash



Instalación y creación del archivo

1. Asegurarte que tienes instalado Python
2. Instalar Dash

pip install dash

3. Crear archivo .py

app.py

Sintaxis

En app.py importamos el paquete

```
1 from dash import Dash, html
```

Inicializar el aplicativo

```
3 app = Dash(__name__)
```

App.layout representa los componentes de aplicación

```
5 app.layout = html.Div([  
6 | |   html.Div(children='Hello World')  
7 | ])
```

Ejecución

```
if __name__ == '__main__':  
| |   app.run(debug=True)
```

Sintaxis

```
1 import dash
2 import dash_core_components as dcc
3 import dash_html_components as html
4 import numpy as np
```

Componente dcc para elementos interactivos

Componente html para el diseño de la página web

```

8 app.layout = html.Div(children=[
9     html.H1(children='Random Scatter Plot'),
10
11     dcc.Graph(
12         id='scatter-plot',
13         figure={
14             'data': [
15                 {
16                     'x': np.random.rand(100),
17                     'y': np.random.rand(100),
18                     'mode': 'markers',
19                     'marker': {
20                         'size': 10,
21                         'color': 'rgb(0, 128, 0)',
22                         'opacity': 0.7
23                     },
24                     'type': 'scatter'
25                 }
26             ],
27             'layout': {
28                 'title': 'Random Scatter Plot'
29             }
30         }
31     )
32 ])

```

Título

Gráfico de dispersión

Datos generados de manera
aleatoria

Conexión a la base de datos



Conexión a la base de datos externa

Hay muchas formas de agregar datos, en diferentes formatos (.txt, json, csv, ..)

Leer datos de un github con pandas

```
3 import pandas as pd
4
5 # Incorporate data
6 df = pd.read_csv(https://raw.githubusercontent.com/plotly/datasets/master/gapminder2007.csv)
7
```


Conexión a la base de datos externa

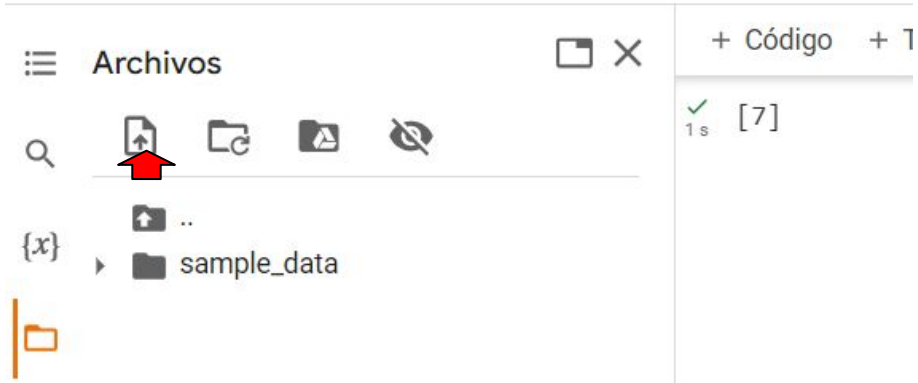
Si tienes los datos de Drive

Leer datos de un drive con pandas

```
6 # Monta tu unidad de Google Drive
7 drive.mount('/content/drive')
8
9 # Ruta al archivo CSV en tu Google Drive
10 file_path = '/content/drive/My Drive/tu_carpeta/tu_archivo.csv'
11
12 # Lee el archivo CSV en un DataFrame
13 df = pd.read_csv(file_path)
```

Si no tienes los datos de Drive

Subir un conjunto de datos



Leer datos de un drive con pandas

```
12 # Lee el archivo CSV en un DataFrame
13 df = pd.read_csv('/content/tu_archivo.csv')
```

Construcción de un aplicativo



```
12 # App layout
13 app.layout = html.Div([
14     html.Div(children='My First App with Data and a Graph'),
15     dash_table.DataTable(data=df.to_dict('records'), page_size=10),
16     dcc.Graph(figure=px.histogram(df, x='continent', y='lifeExp', histfunc='avg'))
17 ])
```

Lectura de datos

Visualización en histograma

Agregando controles a la aplicación

Se realiza a través de la función **callback**, para que la aplicación no sea estático

```
12 # App layout
13 app.layout = html.Div([
14     html.Div(children='My First App with Data, Graph, and Controls'),
15     html.Hr(),
16     dcc.RadioItems(options=['pop', 'lifeExp', 'gdpPercap'], value='lifeExp', id='controls-and-radio-item'),
17     dash_table.DataTable(data=df.to_dict('records'), page_size=6),
18     dcc.Graph(figure={}, id='controls-and-graph')
19 ])
```

```
22 @callback(
23     Output(component_id='controls-and-graph', component_property='figure'),
24     Input(component_id='controls-and-radio-item', component_property='value')
25 )
26 def update_graph(col_chosen):
27     fig = px.histogram(df, x='continent', y=col_chosen, histfunc='avg')
28     return fig
```

Formas de diseñar



Diseño personalizado

Hay varias opciones:

- HTML y CSS
- Componente de Dash Bootstrap
- Componente de Dash Mantine
- El kit de diseño de Dash (DDK)

HTML Y CSS

```
9 # Initialize the app - incorporate css
10 external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
11 app = Dash(__name__, external_stylesheets=external_stylesheets)
12
14 app.layout = html.Div([
15     html.Div(className='row', children='My First App with Data, Graph, and Controls',
16         style={ 'textAlign': 'center', 'color': 'blue', 'fontSize': 30}),
17
18     html.Div(className='row', children=[
19         dcc.RadioItems(options=['pop', 'lifeExp', 'gdpPercap'],
20             value='lifeExp',
21             inline=True,
22             id='my-radio-buttons-final')
23     ]),
```


Dash Bootstrap

Primero instalar **pip install dash-bootstrap-components**

```
15 app.layout = dbc.Container([
16     dbc.Row([
17         html.Div('My First App with Data, Graph, and Controls', className="text-primary text-center fs-3")
18     ]),
19
20     dbc.Row([
21         dbc.RadioItems(options=[{"label": x, "value": x} for x in ['pop', 'lifeExp', 'gdpPercap']],
22                         value='lifeExp',
23                         inline=True,
24                         id='radio-buttons-final')
25     ]),
26
27     dbc.Row([
28         dbc.Col([
29             dash_table.DataTable(data=df.to_dict('records'), page_size=12, style_table={'overflowX': 'auto'})
30         ], width=6),
31
32         dbc.Col([
33             dcc.Graph(figure={}, id='my-first-graph-final')
34         ], width=6),
35     ])
```

Dash Mantine

Primero instalar **pip install dash-mantine-components**

```
15 app.layout = dmc.Container([
16     dmc.Title('My First App with Data, Graph, and Controls', color="blue", size="h3"),
17     dmc.RadioGroup(
18         [dmc.Radio(i, value=i) for i in ['pop', 'lifeExp', 'gdpPercap']],
19         id='my-dmc-radio-item',
20         value='lifeExp',
21         size="sm"
22     ),
23     dmc.Grid([
24         dmc.Col([
25             dash_table.DataTable(data=df.to_dict('records'), page_size=12, style_table={'overflowX': 'auto'})
26         ], span=6),
27         dmc.Col([
28             dcc.Graph(figure={}, id='graph-placeholder')
29         ], span=6),
30     ]),
```

El kit de diseño de Dash (DDK)

No necesita usar HTML o CSS

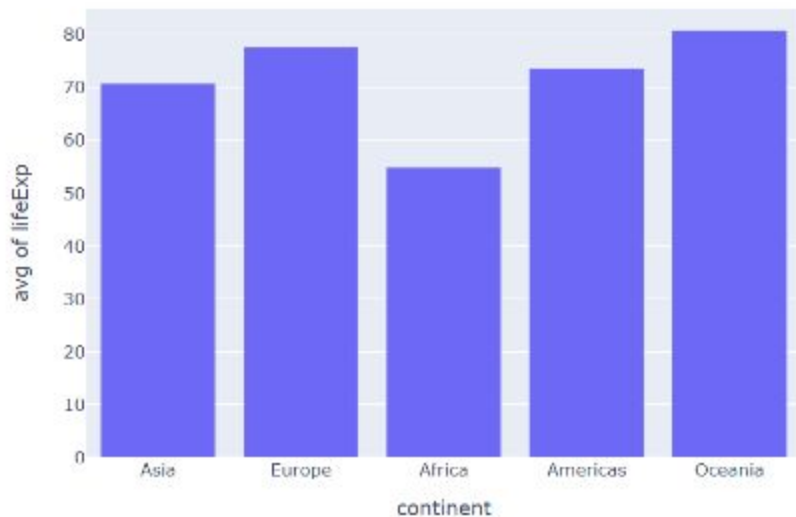
```
14 app.layout = ddk.App([
15     ddk.Header(ddk.Title('My First App with Data, Graph, and Controls')),
16     dcc.RadioItems(options=['pop', 'lifeExp', 'gdpPercap'],
17                     value='lifeExp',
18                     inline=True,
19                     id='my-ddk-radio-items-final'),
20     ddk.Row([
21         ddk.Card([
22             dash_table.DataTable(data=df.to_dict('records'), page_size=12, style_table={'ov
23             ], width=50),
24             ddk.Card([
25                 ddk.Graph(figure={}, id='graph-placeholder-ddk-final')
26             ], width=50),
27     ]),
```

My First App with Data, Graph, and Controls

☐ pop ☒ lifeExp ☐ gdpPercap

country	pop	continent	lifeExp	gdpPercap
Afghanistan	31889923	Asia	43.828	974.5803384
Albania	3600523	Europe	76.423	5937.029525999999
Algeria	33333216	Africa	72.301	6223.367465
Angola	12420476	Africa	42.731	4797.231267
Argentina	40301927	Americas	75.32	12779.37964
Australia	20434176	Oceania	81.235	34435.367439999995
Austria	8199783	Europe	79.829	36126.4927
Bahrain	708573	Asia	75.635	29796.04834
Bangladesh	150448339	Asia	64.062	1391.253792
Belgium	10392226	Europe	79.441	33692.60508
Benin	8078314	Africa	56.728	1441.284873
Bolivia	9119152	Americas	65.554	3822.137084

« < 1 / 12 > »





Gracias!