



Proyecto #1 Aplicaciones con Dash



Equipo de trabajo

21000958

Osiel Gutierrez Herrera

215002659

Jose Andres Salguero Palomo

20023915

Enrique Antonio Viau Najarro



Descripción

Utilizando los CSV, cree un cubo el cual contenga la información de todas las tablas adjuntas, considere el dataset flights como fact table y los datasets airplanes, weather, Airlines como dim tables.

Deberá crear una aplicación la cual permita mostrar gráficas analíticas sobre las entidades relacionadas en los datos. Su aplicación deberá tener las siguientes pestañas/rutas:

- Flights, Airplanes, Airlines, Weather. Las gráficas a realizar son las siguientes:

1.Cantidad de Vuelos en el tiempo (serie temporal).

Dashboard

1.Cantidad de Vuelos en el tiempo

2.Cantidad de vuelos por Airline

3.Distancia recorrida y tiempo en el aire

4.Condiciones climáticas

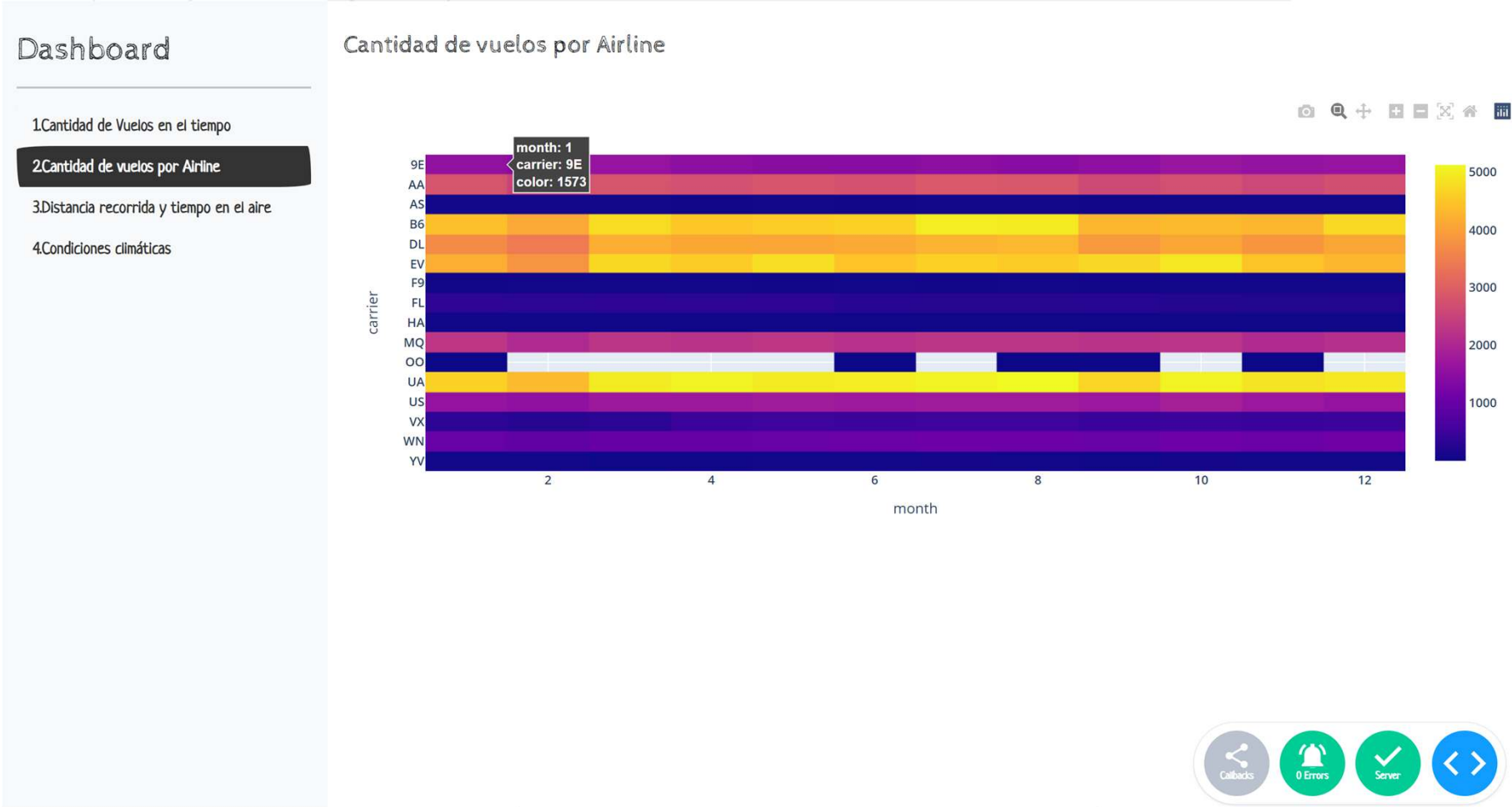
Cantidad de vuelos en el tiempo (serie temporal)



Desarrollo de código

```
pregunta1.py X main.py pregunta2.py pregunta3.py pregunta4.py
pages > pregunta1.py > ...
1 import dash
2 import dash_html_components as html
3 import dash_core_components as dcc
4 import dash_bootstrap_components as dbc
5 import styles
6 from dash.dependencies import Input, Output
7 import plotly.express as px
8 import plotly.graph_objects as go
9
10 import pandas as pd
11 import sqlalchemy
12
13 import config
14 from app import app
15 from app import server
16 from dataframe import df2
17
18
19
20 conteo_vuelos = df2.groupby(["month"], as_index=False)["origin"].count()
21
22 origen = df2['origin'].unique()
23 fig = go.Figure([go.Scatter(x=conteo_vuelos['month'], y=conteo_vuelos['origin'])])
24 fig.update_traces(marker_color='#78c2ad')
25
26
27
28 content = html.Div([html.H4('Cantidad de vuelos en el tiempo (serie temporal)'),
29                       html.Br(),
30                       dcc.Graph(id='time-series-chart', figure=fig)])
31
```

2. Cantidad de vuelos por Airline por mes





Desarrollo de código

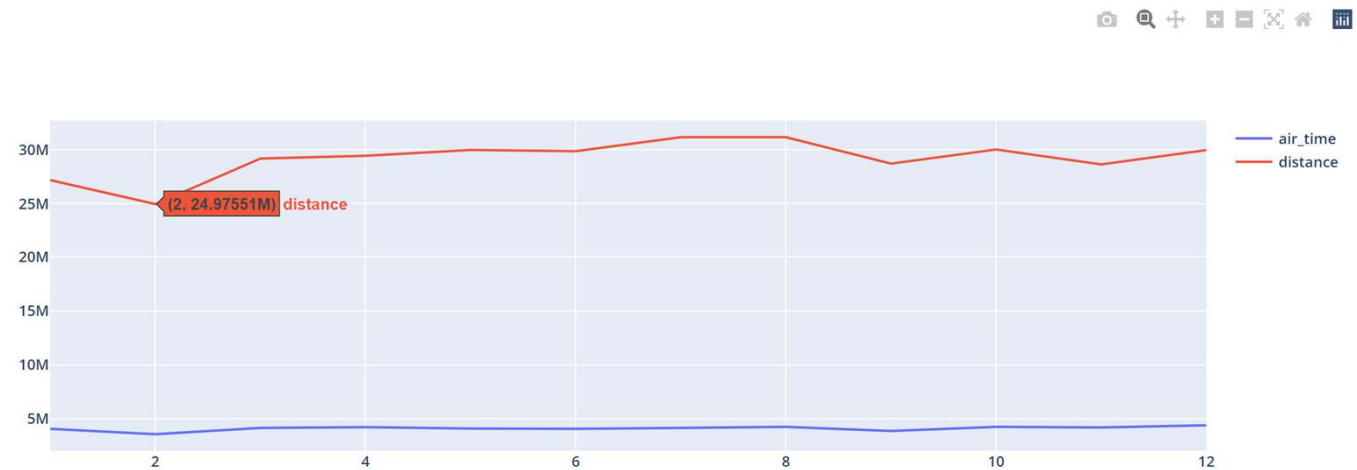
```
pregunta2.py X
pages > pregunta2.py > ...
1 import dash
2 import dash_html_components as html
3 import dash_core_components as dcc
4 import dash_bootstrap_components as dbc
5 import styles
6 from dash.dependencies import Input, Output
7 import plotly.express as px
8 import plotly.graph_objects as go
9
10 import pandas as pd
11 import sqlalchemy
12
13 import config
14 from app import app
15 from app import server
16 from dataframe import df2
17
18
19 df4=pd.DataFrame({'count' : df2.groupby( ['carrier', 'month'] ).size()}).reset_index()
20 df5 = df4.pivot("carrier", "month", "count")
21
22 fig = px.imshow(df5, text_auto=True)
23
24
25
26 content = html.Div([html.H4('Cantidad de vuelos por Airline'),
27 | | | | | html.Br(),
28 | | | | | dcc.Graph(id='graph', figure=fig) ])
29
30
31
```

3. Estadística de distancia recorrida y tiempo en el aire por avión en cada mes.

Dashboard

- 1.Cantidad de Vuelos en el tiempo
- 2.Cantidad de vuelos por Airline
- 3.Distancia recorrida y tiempo en el aire**
- 4.Condiciones climáticas

Distancia recorrida y tiempo en el aire



Desarrollo de código

```
pregunta3.py X
pages > pregunta3.py > [at] at1
1  import dash
2  import dash_html_components as html
3  import dash_core_components as dcc
4  import dash_bootstrap_components as dbc
5  import styles
6  from dash.dependencies import Input, Output
7  import plotly.express as px
8  import plotly.graph_objects as go
9
10 import pandas as pd
11 import sqlalchemy
12
13 import config
14 from app import app
15 from app import server
16 from dataframe import df2
17
18
19 at = df2.groupby(["month"])["air_time"].sum()
20 at = pd.DataFrame(at)
21 at.reset_index(inplace=True)
22
23 at1 = df2.groupby(["month"])["distance"].sum()
24 at1 = pd.DataFrame(at1)
25 at1.reset_index(inplace=True)
26
27 fig = go.Figure()
28 fig.add_traces(go.Scatter( x=at['month'], y=at['air_time'],
29                             mode='lines',
30                             name='air_time',
31                             line_shape='linear'))
32 fig.add_trace(go.Scatter(x=at1['month'], y=at1['distance'],
33                             mode='lines',
34                             name='distance'))
35
36
37 content = html.Div([html.H4('Distancia recorrida y tiempo en el aire'),
38                       html.Br(),
39                       dcc.Graph(id='graph', figure=fig) ])
```

4. Condiciones climáticas: Esta gráfica deberá mostrar en una serie de tiempo la temperatura, humedad relativa y la velocidad del viento, por cada aeropuerto



Desarrollo de código

1

```
pregunta4.py X
pages > pregunta4.py > ...
1 import dash
2 import dash_html_components as html
3 import dash_core_components as dcc
4 import dash_bootstrap_components as dbc
5 import styles
6 from dash.dependencies import Input, Output
7 import plotly.express as px
8 import plotly.graph_objects as go
9
10 import pandas as pd
11 import sqlalchemy
12
13 import config
14 from app import app
15 from app import server
16 from dataframe import df2
17
```

pregunta4.py X

pages > pregunta4.py > ...

```
17
18
19 gf = df2.groupby(["month"])["temp"].sum()
20 gf = pd.DataFrame(gf)
21 gf.reset_index(inplace=True)
22
23
24 gf1 = df2.groupby(["month"])["humid"].sum()
25 gf1 = pd.DataFrame(gf1)
26 gf1.reset_index(inplace=True)
27
28 gf2 = df2.groupby(["month"])["wind_speed"].sum()
29 gf2 = pd.DataFrame(gf2)
30 gf2.reset_index(inplace=True)
31
32 fig = go.Figure()
33 fig.add_traces(go.Scatter( x=gf['month'], y=gf['temp'],
34                             mode='lines',
35                             name='temp',
36                             line_shape='linear'))
37 fig.add_trace(go.Scatter(x=gf1['month'], y=gf1['humid'],
38                             mode='lines',
39                             name='humid'))
40 fig.add_trace(go.Scatter(x=gf2['month'], y=gf2['wind_speed'],
41                             mode='lines',
42                             name='wind_speed'))
43
44
45
46 content = html.Div([html.H4("Condiciones climáticas"),
47                       html.Br(),
48                       dcc.Graph(id='graph', figure=fig) ])
```

2

Recursos utilizados

- VisualCode
- Archivos de Jupyter Notebook para realización de pruebas

Enlace a GitHub

- https://github.com/OsielGH/Proyecto-1_Product_Development-.git

Nombre

- __pycache__
- pages
- 1 - Dash.pdf
- airlines.csv
- airports.csv
- app.py
- config.py
- dataframe.py
- flights.csv
- main.py
- planes.csv
- Proyecto_AplicacionesDash.pdf
- Proyecto_AplicacionesDash.pptx
- styles.py
- weather.csv

Nombre

- __pycache__
- error.py
- pregunta1.py
- pregunta2.py
- pregunta3.py
- pregunta4.py