

In [9]:

```
import pandas as pd

import networkx as nx

import matplotlib.colors as mcolors

import dash
import dash_html_components as html
import dash_cytoscape as cyto
```

Base de Dados

O objetivo final do projeto é usar uma base de dados com as relações de parentesco dentro do bairro de Surubeju do município de Monte Alegre - PA. No entanto, devido ao tempo de coleta desses dados, usarei um apenas uma amostra ou um conjunto de dados fake para a criação da visualização.

Motivação

No referido bairro, no qual eu resido, existe uma peculiaridade de que os habitantes deste local tendem a constituir famílias com moradores do próprio bairro. Diante disso, criaram-se relações de parentesco peculiares, complexas e até mesmo desconhecidas pelos residentes. Com isso, decidi criar uma visualização dinâmica que demonstrasse todas as relações de parentesco presentes no bairro.

Expectativa

O objetivo é conseguir plotar um gráfico de redes interativo, em que o usuário possa inserir o nome de duas pessoas e verificar a rede de parentesco entre elas.

Esboços

Estão algumas amostras de dados com relações de parentesco e esboços de como será a visualização final.

In [10]:

```
#Coletando Data Frame do xls
df = pd.read_excel('Static-visuzalization-test/familia.xls')
df.head()
```

Out[10]:

	filho	pai-mae
0	Nira	Nonato
1	Jovelina	Nonato
2	Josias	Nonato
3	Tiana	Nonato
4	Nonato	Nonato

In [11]:

```
## number of classes
len(df['pai-mae'].unique())
```

Out[11]:

9

In [12]:

```
# Contando o número de pessoas no banco de dados
linhas = len(df['pai-mae'].unique())
# Criando uma lista de cores disponíveis
chosen = list(mcolors.cnames.keys())
# Criando uma lista de cores de acordo com o número de pessoas
chosen = chosen[1:linhas+1]
# Dicionário das pessoas correspondentes a cada cor
colors = dict(zip(df['pai-mae'].unique(), chosen))
print(colors)
# Criando um dataframe com as pessoas e suas cores correspondentes.
df_colors = pd.DataFrame({'classes': df['pai-mae'].unique(), 'colors': chosen})
df_colors.head()
```

```
{'Nonato': 'antiquewhite', 'Jovelina': 'aqua', 'Fernando': 'aquamarine', 'Tiana': 'azure', 'Josias': 'beige', 'Nira': 'bisque', 'Michele': 'black', 'Dani': 'blanchedalmond', 'Terri': 'blue'}
```

Out[12]:

	classes	colors
0	Nonato	antiquewhite
1	Jovelina	aqua
2	Fernando	aquamarine
3	Tiana	azure
4	Josias	beige

In [13]:

```

#Criando entidade do Grafo
G = nx.DiGraph()
# Adicionando Nós
# Nessa etapa são adicionados os nós soltos com valores e cores
for idx in df.index:
    G.add_node(
        df.loc[idx, 'filho'],
        label=df.loc[idx, 'filho'],
        color=colors[df.loc[idx, 'pai-mae']],
        className=df.loc[idx, 'filho']
    )

# Nessa etapa, são estabelecidas as conexões entre os grafos
G.add_edges_from(zip(df['filho'].values, df['pai-mae'].values),
                 color='turquoise',
                 relation = 'filho de')

for g in G.nodes():
    G.edges(data=True)

```

In [14]:

```

cyto_family_nodes = nx.readwrite.json_graph.cytoscape_data(G)['elements']['nodes']
cyto_family_edges = nx.readwrite.json_graph.cytoscape_data(G)['elements']['edges']

```

In []:

```

app = dash.Dash("Family Network")
app.layout = html.Div([
    html.P('Dash Cytoscape:'),
    cyto.Cytoscape(
        id='cytoscape',
        elements=cyto_family_nodes+cyto_family_edges,
        layout={
            'name':'breadthfirst',
            # 'roots': ' #Terri #Nonato'
            'roots': '#Nonato'
        },
        style={'width':'400px', 'height':'500px'},
        stylesheet=[
            {'selector': 'node',
             'style': { 'label': 'data(label)',
                        "text-wrap": "wrap"}
            ]
    )
])
app.run_server(debug=False)

```

Dash Cytoscape:

