



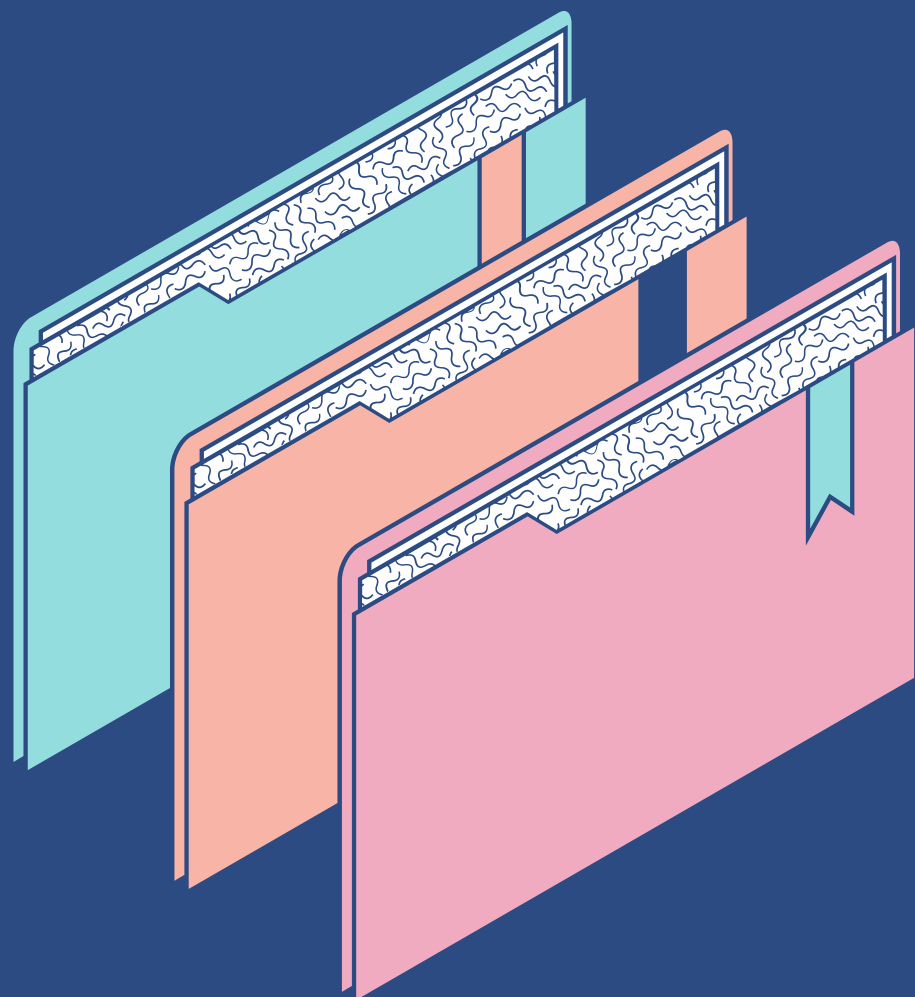
:hiper  
stream

# Desafio: Desenho de fluxo de aplicações

Equipe Conexão

Integrantes:

- Amanda Pageú Silva
- Gabriel Akira Wakavaiachi
- Renan Yudi Fukumori

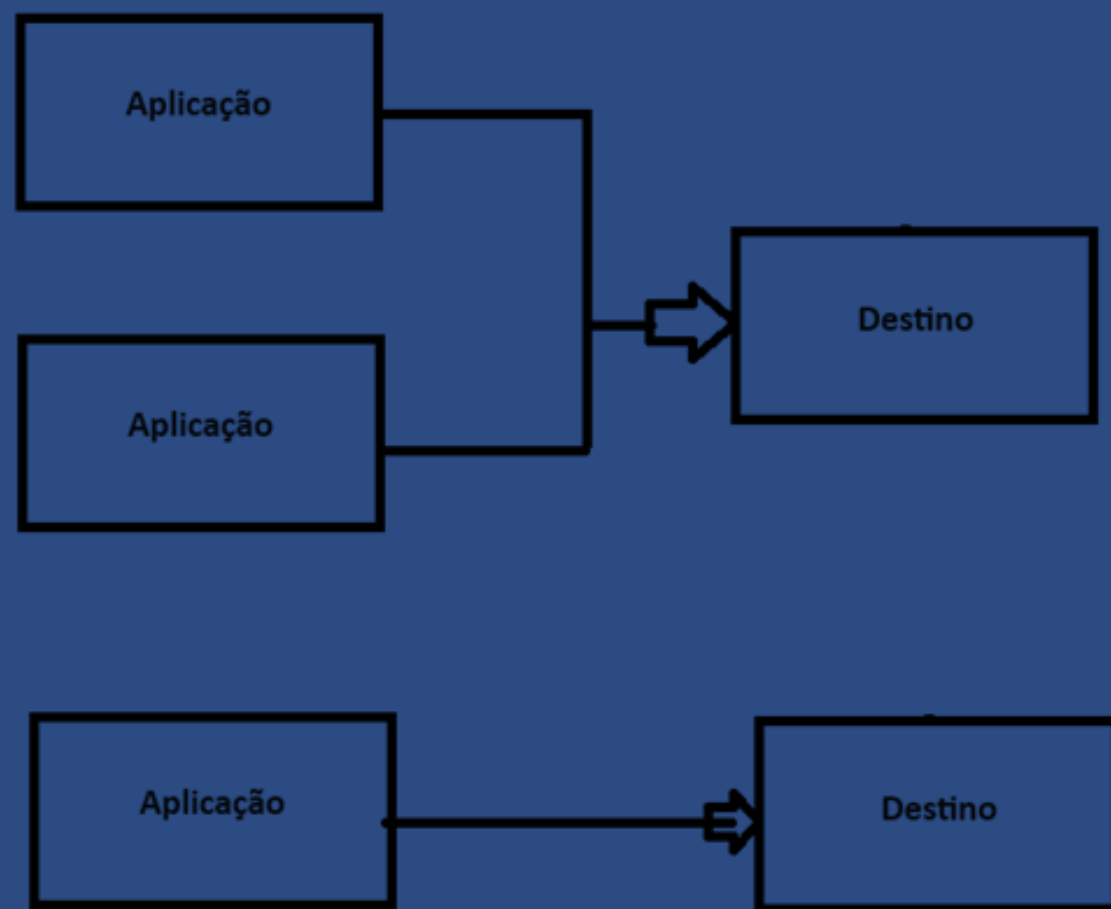


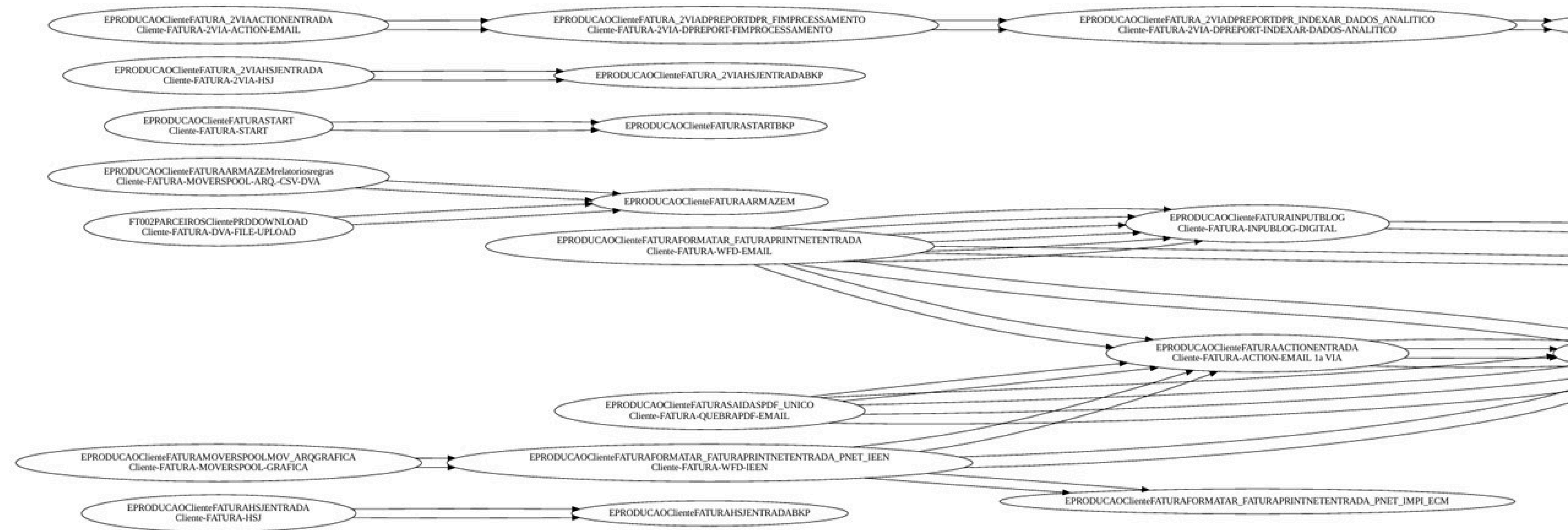
# Construção do Código

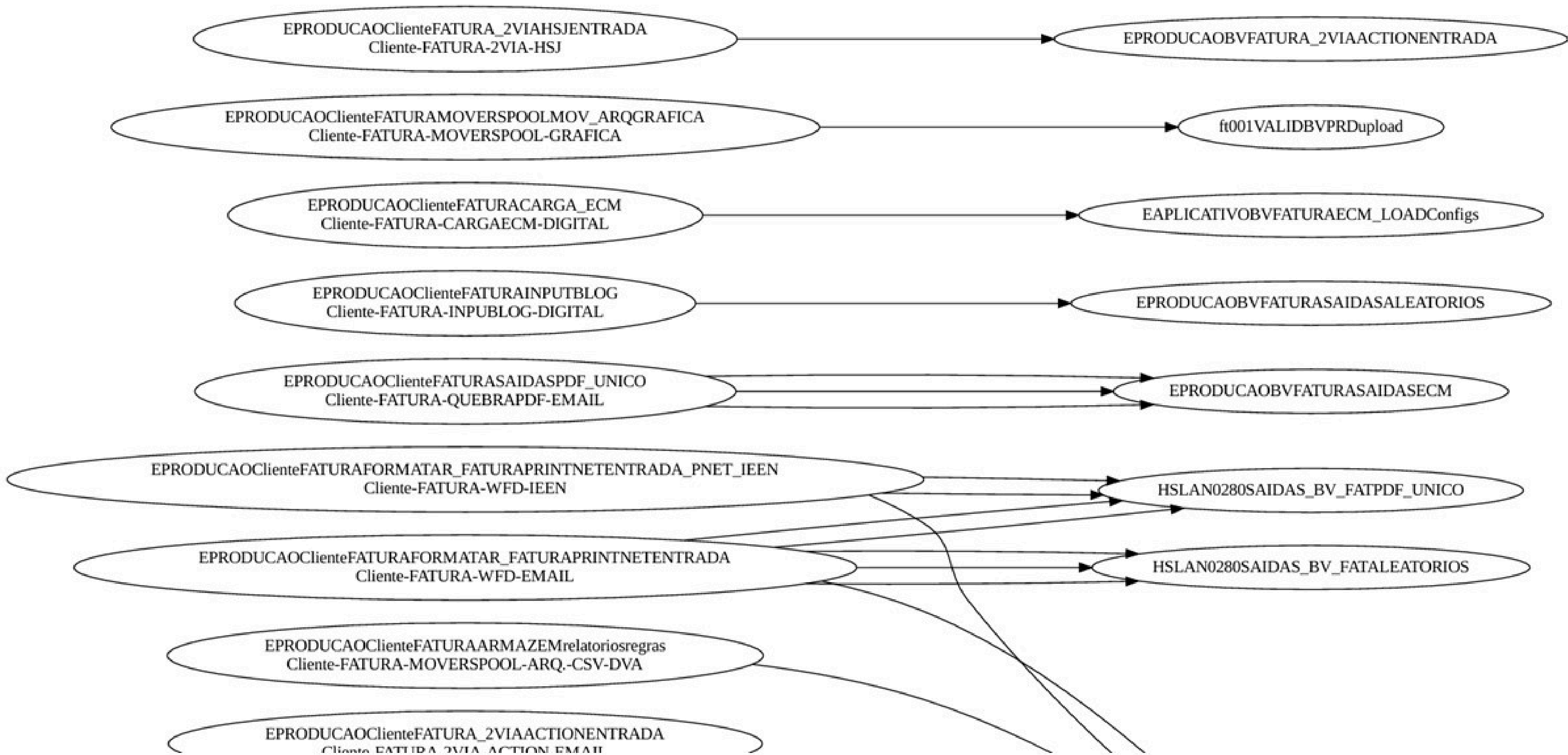
COMO CONSTRUÍMOS E O QUE PENSAMOS:

Para nossa resolução nós utilizamos o recurso Collaboratory do Google, utilizando a linguagem python. De início, visualizamos a base de dados e, em um primeiro momento, cogitamos que poderia ser observado de três formas distintas.

Por meio dos códigos criamos dois diagramas, um demonstrando as movimentações dos arquivos de uma pasta origem para uma pasta destino e um outro demonstrando a movimentação da pasta origem para a pasta backup.







```
import graphviz as gz
import pandas as pd
import re

url = "https://raw.githubusercontent.com/gabakiwa/Hackathon/main/baseparatestes-1.csv"
Base = pd.read_csv(url)

grafo = {}
dot = gz.Digraph(graph_attr={'rankdir': 'LR'})
for _, row in Base.iterrows():
    aplicacao = re.sub(r"^\w:/-", "", row["Nome"])
    origem = re.sub(r"\W", "", row["PastaOrigem"])
    destino = re.sub(r"\W", "", row["PastaDestino"]) if not pd.isnull(row['PastaDestino']) else ""

    if backup is not None:

        if origem not in grafo:
            grafo[origem] = []

            label = f"{origem}\n{row['Nome']}"
            dot.node(origem, label=label)

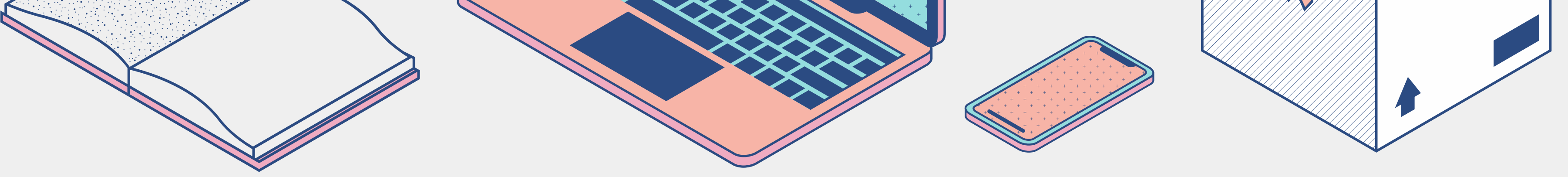
            grafo[origem].append(destino)

            dot.node(destino)

for k, v in grafo.items():
    for n in v:
        dot.edge(k, n)

dot.attr(rank='same')
for i, node in enumerate(grafo.keys()):
    dot.attr(rank='same', _attributes={'rank': 'same', 'same': f'{node};;'})

dot.render('OrigemDestino', format='pdf', cleanup=True)
print("Diagrama do grafo dividido em camadas verticais gerado com sucesso como 'OrigemDestino.pdf'")
```



# Considerações Finais

- Praticidade e objetividade;
- Fácil compreensão;
- Nitidez e fluidez das informações;
- Versatilidade.