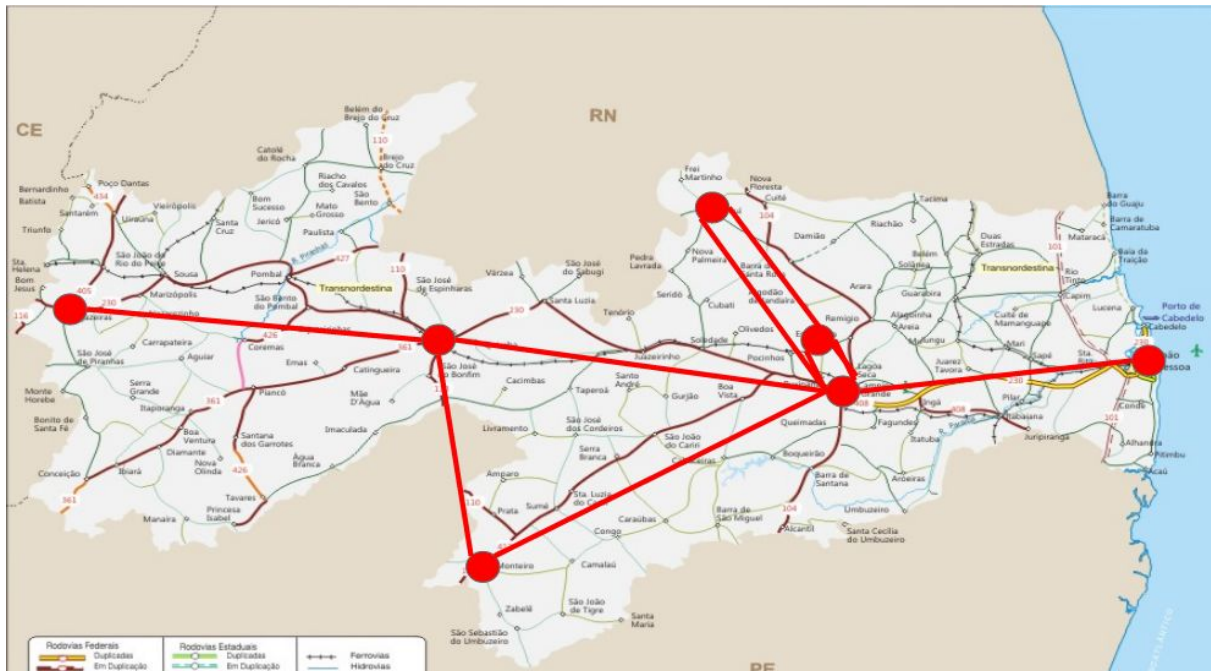


Roteiro 1



No exemplo acima, os vértices representam cidades e as arestas indicam se é possível chegar a outra cidade por uma estrada.

Dessa forma, esse grafo pode ser escrito da seguinte forma:

$N = \{J, C, E, P, M, T, Z\}$

$A = \{a1, a2, a3, a4, a5, a6, a7, a8, a9\}$

$g(a1) = JC, g(a2) = CE, g(a3) = CE, g(a4) = CP, g(a5) = CP, g(a6) = CM, g(a7) = CT, g(a8) = MT, g(a9) = TZ$

1. Construa o grafo da Paraíba usando o módulo `grafo.py` disponibilizado [aqui](#) e o imprima na saída padrão. Use `import` para incluir `grafo.py` em seu próprio módulo:

```
from grafo import Grafo
```

obs.: Você não deve fazer quaisquer modificações em `grafo.py`

2. Faça um programa que receba um grafo a partir da entrada padrão e o imprima na saída padrão. O formato de entrada é o seguinte:
 - a. A primeira linha contém todos os vértices, separados por vírgula e espaço.
Ex.:
J, C, E, P, M, T, Z
 - b. Os vértices não podem incluir os caracteres "-", "(", ")" e "
 - c. A segunda linha contém os nomes das arestas seguido dos vértices que ela conecta. Siga o seguinte formato:
a1(J-C), a2(C-E), a3(C-E), ...
 - d. O nome de cada vértice pode ter um tamanho arbitrário de caracteres

- e. O programa deve funcionar para qualquer grafo informado pelo usuário
 - f. Você deve tratar problemas com a formatação da entrada
3. Crie funções em Python para satisfazer os seguintes questionamentos:
- a. Encontre todos os pares de vértices não adjacentes.
 - b. Há algum vértice adjacente a ele mesmo? (Retorne True ou False)
 - c. Há arestas paralelas? (Retorne True ou False)
 - d. Qual o grau do vértice C (Faça uma função genérica para calcular o grau de qualquer vértice. Em seguida, use-a para verificar o grau do vértice C)
 - e. Quais arestas incidem sobre o vértice M?
 - f. Esse grafo é completo?
 - g. (DESAFIO) Encontre um ciclo, se houver (Retorne a sequência de vértices e arestas do ciclo ou False se não houver ciclo)
 - h. (DESAFIO) Encontre um caminho de comprimento 4, se houver (Faça uma função genérica que encontre um caminho de tamanho arbitrário)
 - i. (DESAFIO) Esse grafo é conexo?