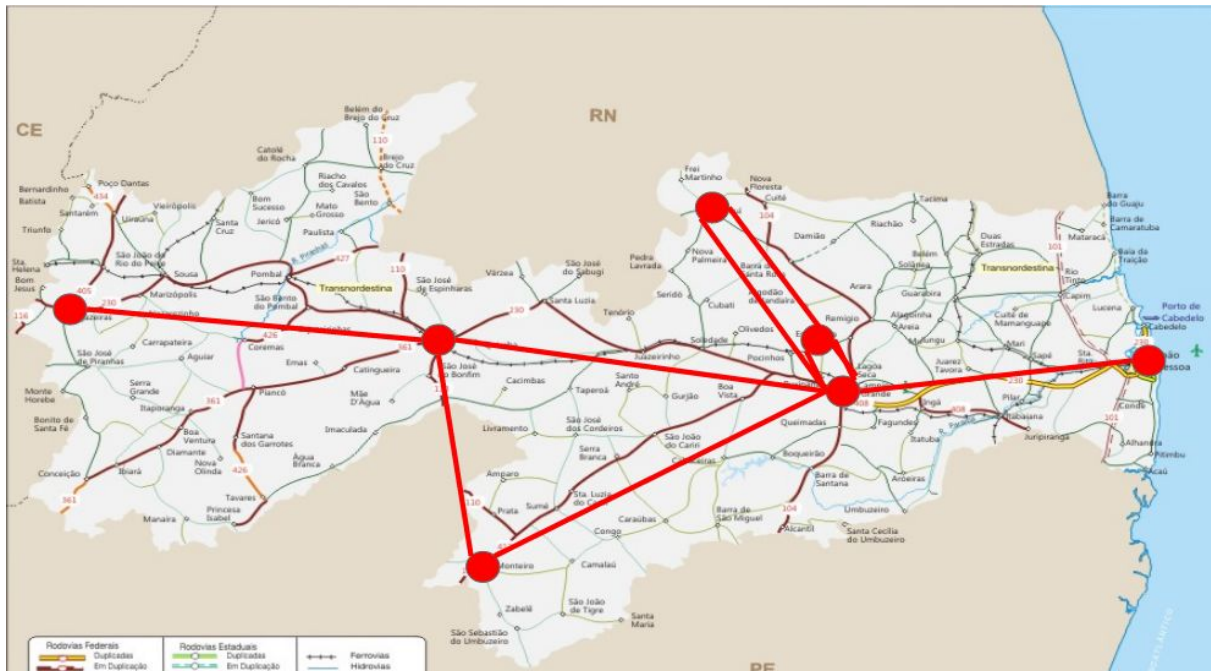


Roteiro 1



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
```

2. 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 de um vértice arbitrário?
 - e. Quais arestas incidem sobre um vértice N arbitrário?
 - 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 n (n passado como parâmetro), se houver.

- i. (DESAFIO) Esse grafo é conexo?
- j. Para essa atividade foi criado um [conjunto de casos de teste](#). Use-o para testar seu módulo em Python.
- k. As funções devem ser criadas dentro do arquivo grafo.py