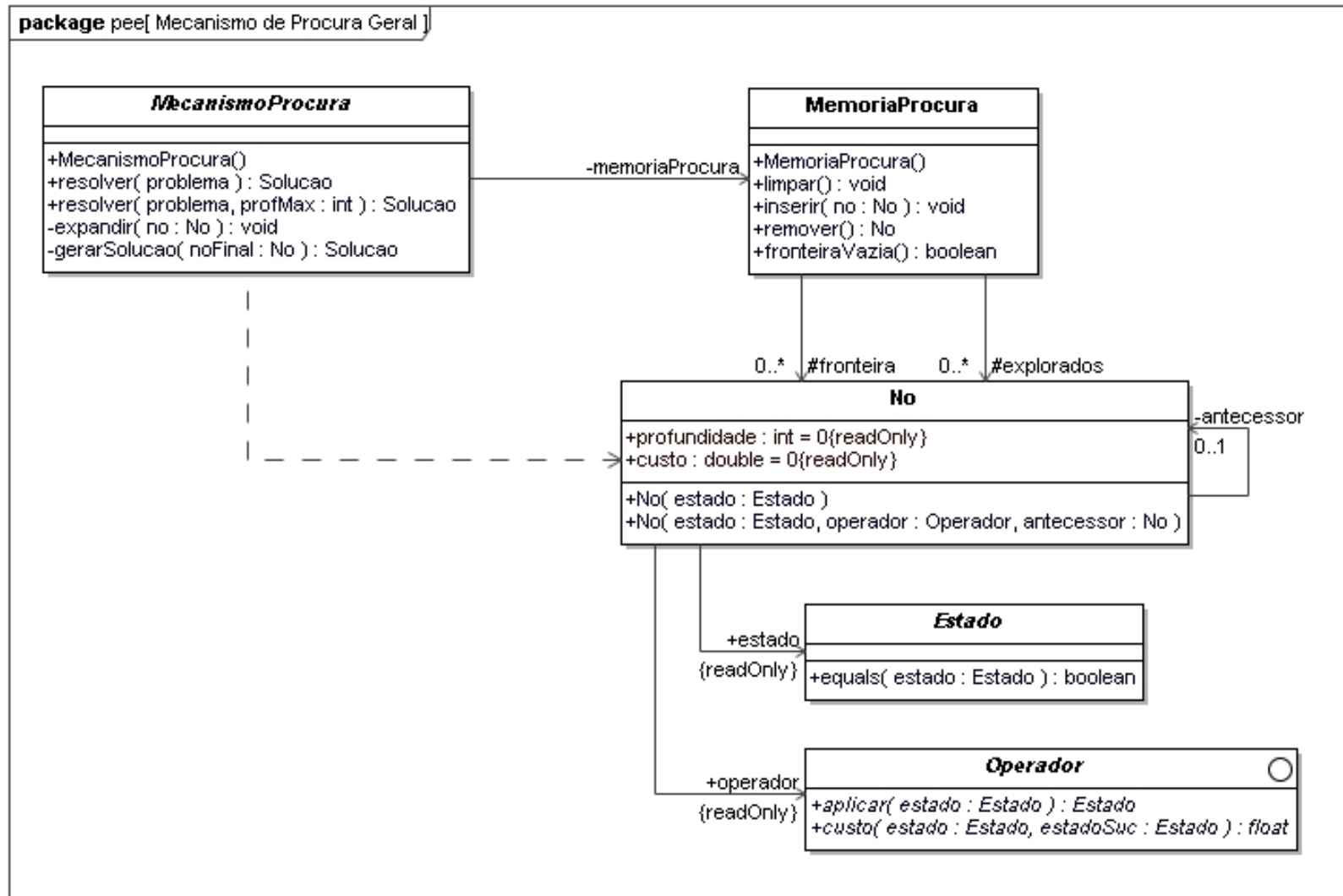
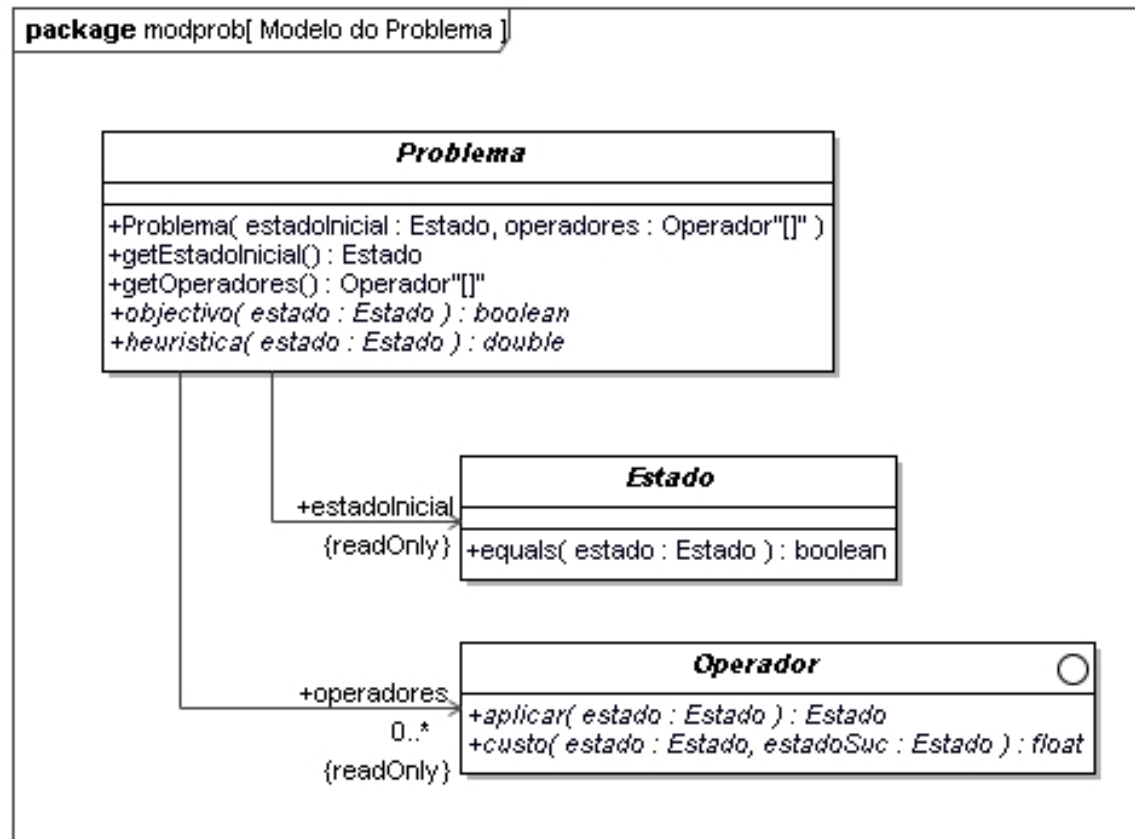


# MECANISMO DE PROCURA GERAL



# MODELO DO PROBLEMA



# RESOLVER PROBLEMA

Com limitação de profundidade

```
function resolver(problema : Problema, profMax : int) : Solucao
1.  memoriaProcura.limpar()
2.  noInicial = No(problema.estadoInicial)
3.  memoriaProcura.inserir(noInicial)
4.  while not memoriaProcura.frenteiraVazia():
5.      no = memoriaProcura.remove()
6.      if problema.objectivo(no.estado):
7.          return gerarSolucao(no)
8.      else:
9.          if no.profundidade < profMax):
10.             expandir(no)
```

Sem limitação de profundidade

```
function resolver(problema : Problema) : Solucao
1.  resolver(problema,  $\infty$ )
```

# EXPANDIR NÓ

```
function expandir(no : No)
1.  estado = no.estado
2.  for operador in problema.operadores:
3.      estadoSuc = operador.aplicar(estado)
4.      if estadoSuc is not None:
5.          noSuc = No(estadoSuc, operador, no)
6.          memoriaProcura.inserir(noSuc)
```