



```

class ControladorEntidades {
    public ??? findVizinhoDe(??? c, ??? e) {
        ??? v[] = new ???[10];
        Construc o h = c;
        if(/* omitido */) {
            Ferraria f = (Ferraria)h;
            v[0] = f;
        }
        if(/* omitido */) {
            Infantaria i = (Infantaria)e;
            v[1] = e;
        }
        Unidade u = e;
        v[2] = u;
        v[3] = c;
        ??? z = (Arqueiro) u;
        return v;
    }
}

```

1 = *Entidade[]* ou *Atualizavel[]*. S o as classes/interfaces mais abstratas implementadas por todas as subclasses, e podem retornar objetos de qualquer subclasse.

2 = *Construc o*. *Construc o h* aponta para *c*, e em seguida *Ferraria f* faz Downcasting de *h*, sendo necess rio que *c* seja um objeto de classe superior a *Ferraria*.

3 = *Unidade*. *Unidade u* aponta para *e*. Antes disso *Infantaria i* fez Downcasting de *e*, sendo necess rio que *e* seja um objeto de classe superior a *Infantaria*.

4 = *Entidade* ou *Atualizavel*. Ser o inseridos objetos das classe *Unidade* e *Construc o* e suas filhas nesse vetor, portanto ele precisa ser de uma superclasse de ambas para aceit -los.

5 = *Arqueiro*. A v riavel *z* est  fazendo Downcasting de *u*. Isso significa que *z* deseja utilizar o objeto *u* como *Arqueiro*.

OBS: 1 e 4 devem ser da mesma classe.