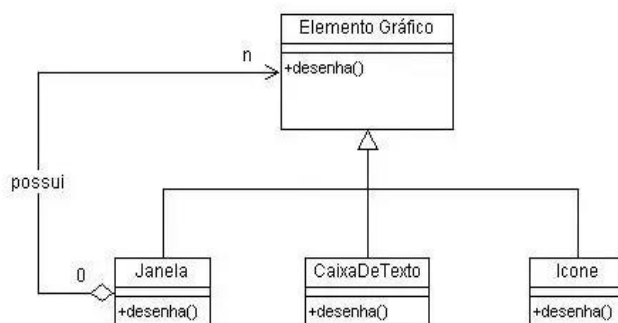


## Padrões Iterator e Composite

O padrão *Composite* é muito utilizado com o *Iterator* para percorrer estruturas em forma de árvore. A ideia desse Padrão *Composite* é montar uma árvore onde tanto folhas (objetos individuais) quanto os compostos (grupos de objetos) estejam no mesmo patamar. Em orientação a objetos, isso significa que aplicaremos polimorfismo para chamar métodos de um objeto na árvore sem nos preocuparmos se ele é uma folha ou um composto. Um exemplo seria uma página web, onde o mesmo contém ícones, caixas de texto, janela, etc, todos esses elementos são considerados elementos gráficos, inclusive a página. Levando em conta que há uma hierarquia entre os componentes, indicando o elemento gráfico como uma super-classe, comum entre todas que representam elementos gráficos.



O Padrão *Iterator* tem como finalidade encapsular uma iteração, no entanto ele depende de uma interface chamada *Iterator*. O método *hasNext()* observa se existe mais elementos para serem iterados, obtendo um retorno do tipo boolean. Já o método *next()* retorna o próximo objeto da iteração. Prover acesso a sua representação sem expor muito detalhamento.

