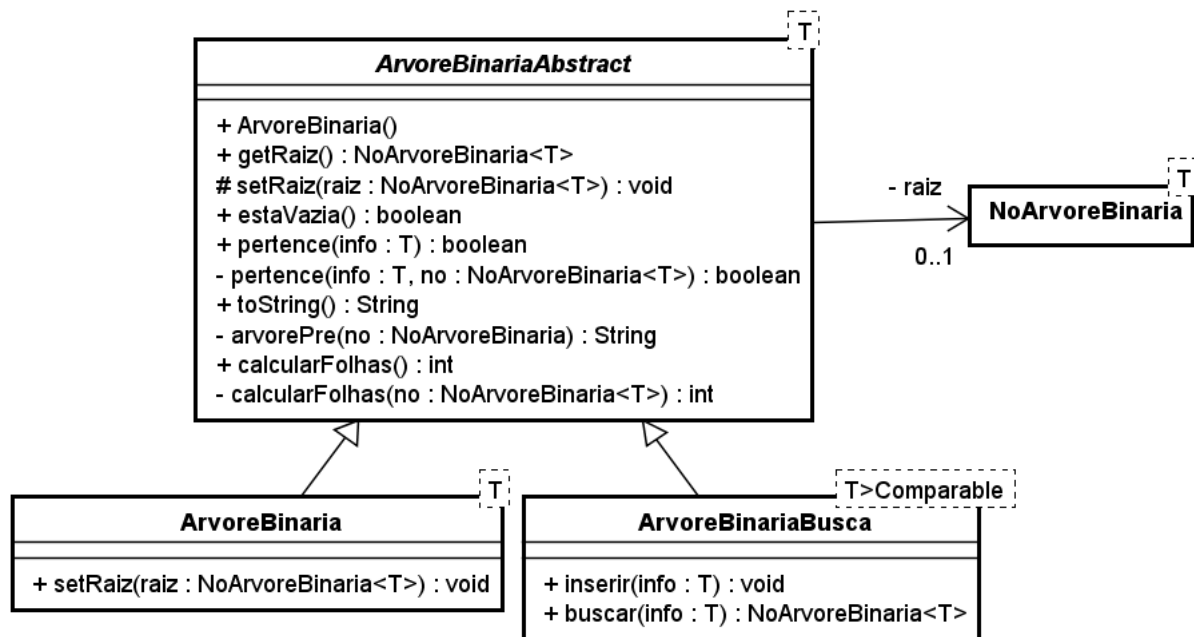


### Lista de Exercício 08

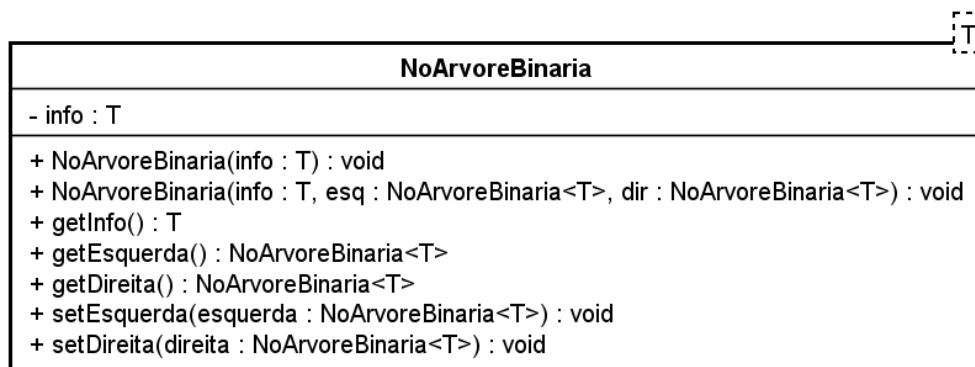
O objetivo desta atividade prática é realizar a implementação de árvores binárias de busca, de acordo com o diagrama de classes da figura seguinte.

Para construir a solução, recomenda-se aproveitar a implementação do exercício 7, como descrito nestas etapas:

- Copiar as classes **ArvoreBinaria** e **NoArvoreBinaria** do exercício anterior para um novo projeto;
- Renomear a classe **ArvoreBinaria** para **ArvoreBinariaAbstract**;
- Tornar a nova classe **ArvoreBinariaAbstract** uma classe abstrata;
- Criar uma nova classe **ArvoreBinaria** estendendo-a da classe **ArvoreBinariaAbstract**;
- Tornar o método **setRaiz()**, da classe **ArvoreBinariaAbstract**, protegido;
- Inserir o método **setRaiz()** na classe **ArvoreBinaria** e implementar seu código para utilizar a implementação do método **setRaiz()** da super classe.



A classe **NoArvoreBinaria** deste exercício difere do projeto do exercício 7, pois deve-se acrescentar *setters* para as variáveis esquerda e direita, como segue abaixo:



Os métodos novos para serem implementados na classe **ArvoreBinariaBusca** são:

- inserir()**: este método deve inserir o dado, fornecido como argumento, na árvore binária de busca.
- buscar()**: este método deve buscar o dado fornecido como argumento, na árvore binária, retornando o nó que o armazena.

Já na classe **NoArvoreBinariaAbstract**, inclua o método **calcularFolhas()**, que deverá retornar a quantidade de nós externos (folhas) na árvore binária.

Após implementar a árvore binária, crie uma classe contendo o método **main()** para testar e demonstrar o funcionamento da árvore implementada.