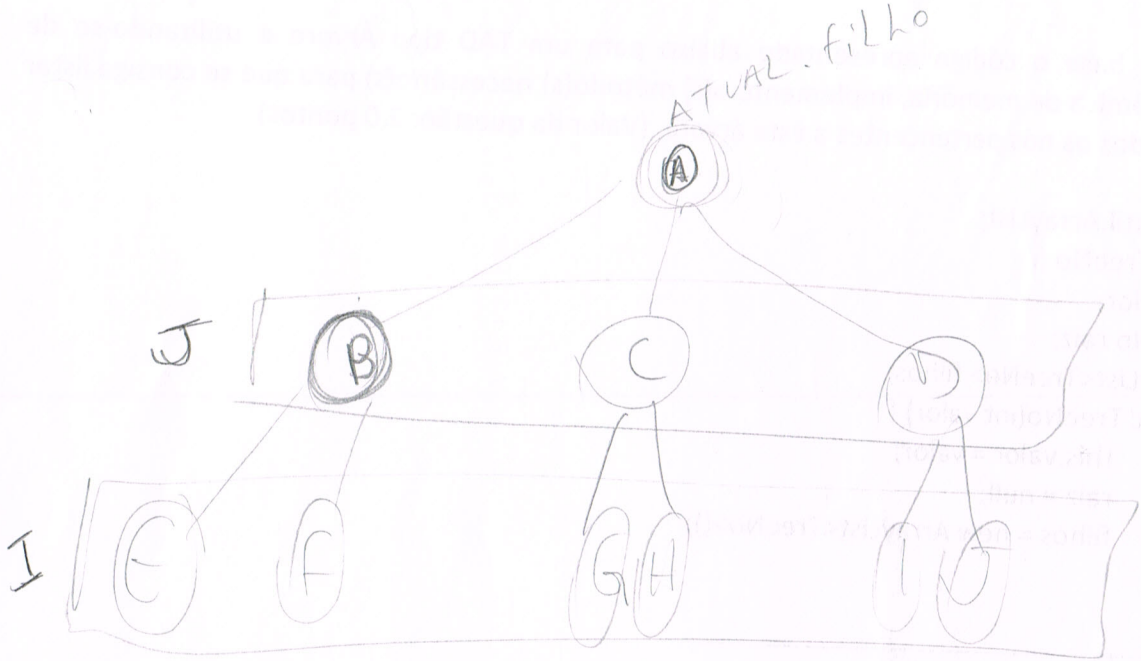


ATUAL = PRIMEIRO

```
for(int j=0; j<atual.filho().size(); j++){
  for(int i=0; i<atual.filho(i).filhos.size(); i++)
      sys
```

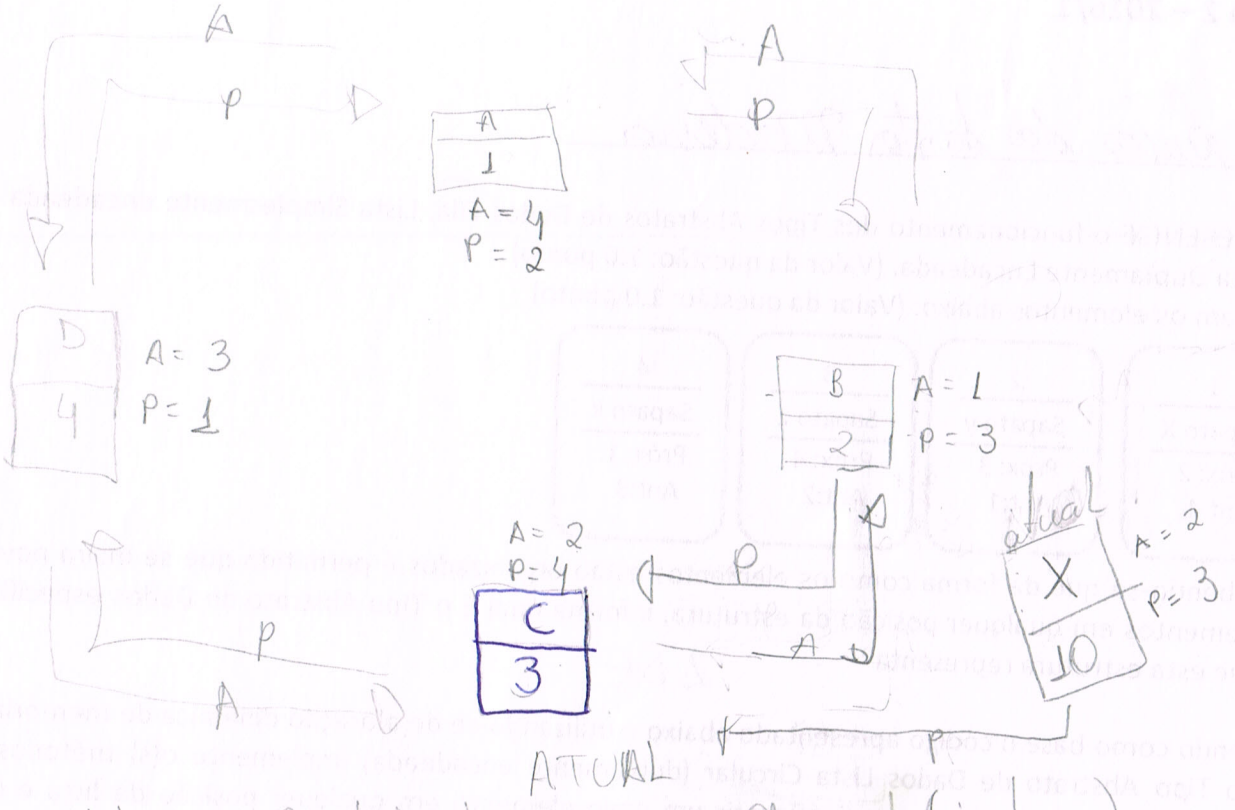
```
for(int i=0; i<raiz.size(); i++){
  sys(atual.filhos()) Lista toda
    atual = atual.filhos(i);
```

① Fila → primeiro

ATUAL → A B C

PRIMEIRO → A

ÚLTIMO → D




```
public No Projeto Dupla posicionaElemento(int n)
ATUAL = PRIMEIRO
for(int i = 1; i < n; i++) {
  ATUAL = ATUAL.PROXIMO;
  i++;
}
```

① atual.anterior.proximo = novo;

atual.anterior = novo;

novo.proximo = atual;

② atual.anterior.proximo = atualproximo;

③ atual.anterior = null;

② atual.proximo.anterior = atual.anterior;

④ atual.proximo = null;

i = 1 2 3

1 2 3

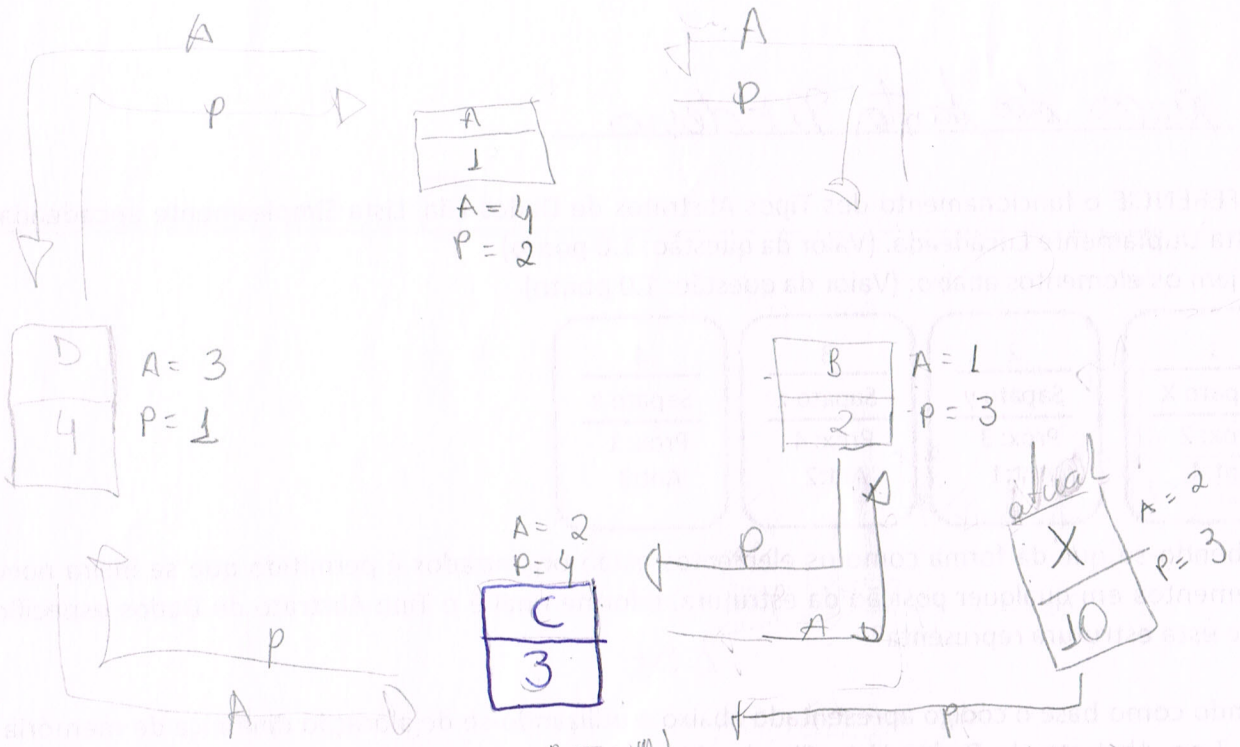
A B C

① Fila → primeiro

ATUAL → A B C

PRIMEIRO → A

ÚLTIMO → D



public No Projeto Dupla posicionaElemento(int n)

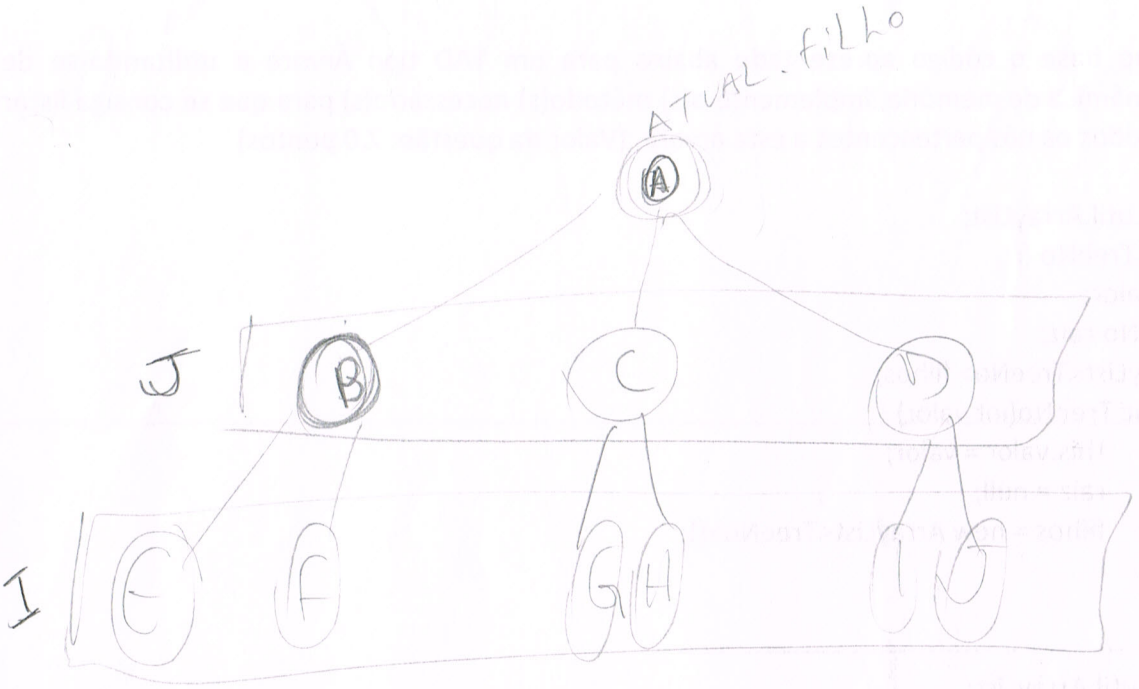
```
ATUAL = PRIMEIRO
FOR(int i = 1 ; i < n ; i++) {
    ATUAL = ATUAL.PROXIMO;
    i++;
}
```

① atual.anterior.proximo = novo;
atual.anterior = novo;
novo.proximo = atual;

② atual.anterior.proximo = atual.proximo;
③ atual.anterior = null;
④ atual.proximo = null;

i = ~~12~~ 3

1 2 3
A B C



ATUAL = PRIMEIRO

```
for(int j=0; j<atual.filho(i).filhos.size(); j++){
  for(int i=0; i<atual.filho(i).filhos.size(); i++)
```

sys

```
for(int i=0; i<raiz.size(); i++){
  sys(atual.filhos(i)); Lista toda
    atual = atual.filhos(i);
```

Lucas dos Santos Moura

4.0

① Fila → O primeiro que entra, é o primeiro a sair. Possui um indicador que mostra o primeiro, o último e o próximo.

Lista Simples → Pode ser retirado e adicionado elemento em qualquer posição da lista. Possui o indicador do próximo elemento e do elemento atual. Além do primeiro e último

Lista Dupla → Pode ser retirado e adicionado elemento em qualquer posição da lista. Possui o indicador do proximo elemento, do anterior e do elemento atual. Além do primeiro e último.

② Lista Duplamente Encadeada.

③
```
public NoProjetoDupla posicionaElemento(int n){
        atual = primeiro;
        for(int i=0; i<n; i++){
            atual = atual.proximo;
        i++;
        }
    return atual;
}

public void insereVariado(NoProjetoDupla novo; int p){

    if((primeiro == null){
        primeiro = novo;
        ultimo = novo;
        atual = novo;
        novo.proximo = null;
        novo.anterior = null;
    }else{
```

```
        posicionaElemento(p);
        novo.anterior = atual.anterior;
        novo.proximo = atual;
        atual.anterior.proximo = novo;
        atual.anterior = novo;
        }
    }

public void excluiVariado(NoProjetoDupla no, int p){
    if(primeiro == null){
        system.out.print("Lista Vazia!");
        }else{
        posicionaElemento(p);
        atual.anterior.proximo = atual.proximo;
        atual.proximo.anterior = atual.anterior;
        atual.proximo = null;
        atual.anterior = null;
        }
    }
```

```
(4) public void listaArvore(){
        atual = primeiro;
    for (int i=0; i<raiz.size(); i++){
        system.out.print(atual + "-" + atual.filhos(i));
        atual = atual.filhos(i);
        i++;
        }
    }
```