

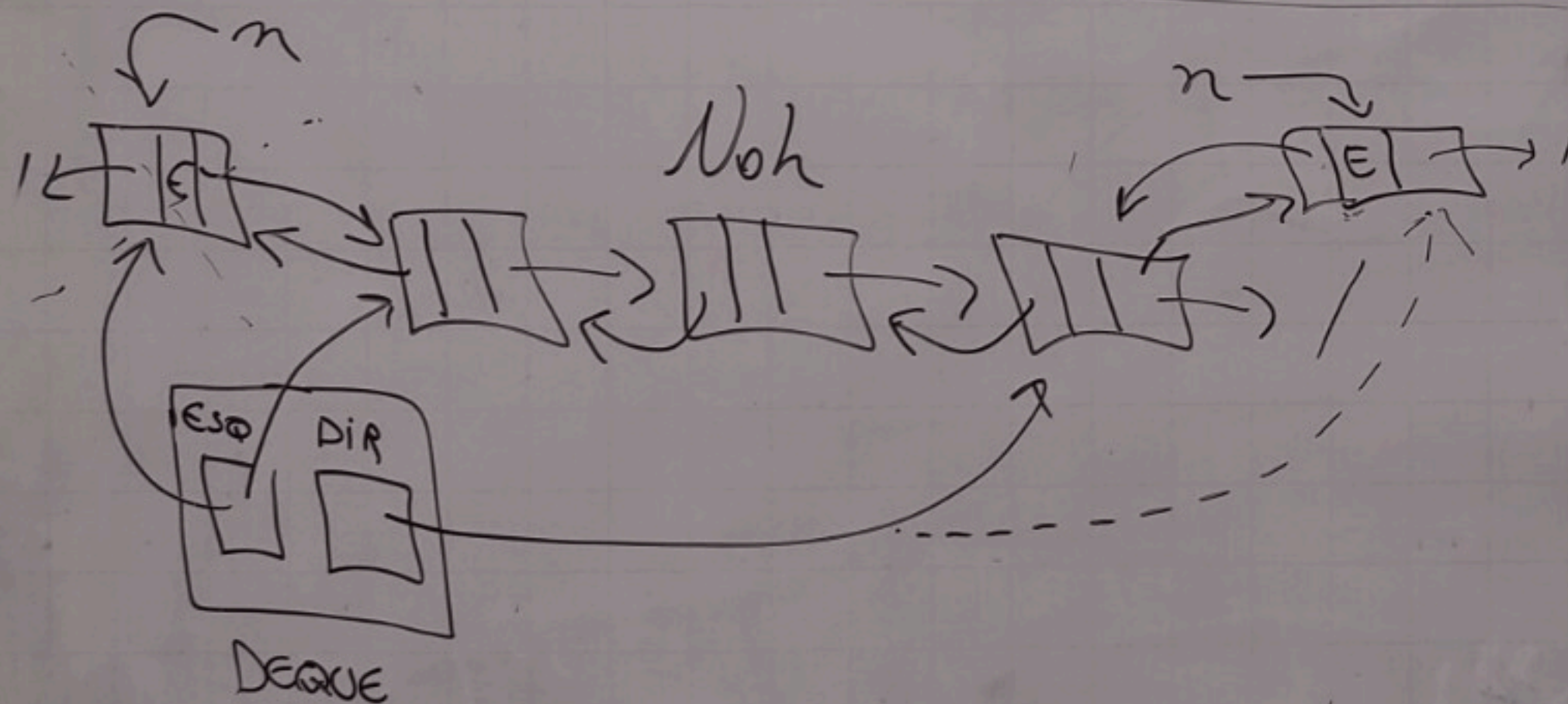
UNIVERSIDADE FEDERAL DO CEARÁ

ESTRUTURAS DE DADOS - TIO140 2022.2 T02A

AULA/ENCONTRO 18 - 17/10/2022

DEQUES (CONTINUAÇÃO)

1. REPRESENTAÇÃO DE DEQUE VIA LISTA ENCADEADA:



2. QUALIFICADORES DE ACESSO EM C++:

- a) PUBLIC: É LIVREMENTE ACESSÍVEL FORA DA CLASSE/ESTRUTURA.
- b) PRIVATE: É ACESSÍVEL SOMENTE DENTRO DA CLASSE/ESTRUTURA.
- c) PROTECTED: É ACESSÍVEL NA CLASSE ATUAL E EM SUAS SUBCLASSES (NÃO USAREMOS).

EXEMPLOS:

a)

```
STRUCT E1
{
    INT i;
    PRIVATE:
    DOUBLE d;
}
```

...
E1 E;
E.i = 2; //ok: PÚBLICO
E.d = 3.14; //ERRO: PRIV.

b) CLASS C1

```
{
  INT i;
  PUBLIC:
  DOUBLE d;
  PRIVATE:
  DOUBLE d2;
  PUBLIC:
  INT i2;
}
```

...
C1 c;
c.i = 3; //ERRO: PRIVADO
c.d2 = 3.1; //IDEM
c.d = 3.1; //OK: PÚBLICO
c.i2 = 4; //IDEM

CONFORME ILUSTRADO ACIMA:

1. POR PADRÃO, OS MEMBROS DE UMA "STRUCT" SÃO PÚBLICOS.
2. POR PADRÃO, OS MEMBROS DE UMA "CLASS" SÃO PRIVADOS.
3. ESSA É A DIFERENÇA ENTRE "STRUCT" E "CLASS" EM C++.
4. QUALIFICADORES DE ACESSO PODEM SER INSERIDOS LIVREMENTE, INCLUSIVE DE FORMA REPETIDA, NUMA CLASSE OU ESTRUTURA.

3. Uma implementação de Deque via Lista Dupl. Encad. 14 5

CLASS DequeInt

{

STRUCT Noh { INT ELEM; Noh *ANT, *PROX; }

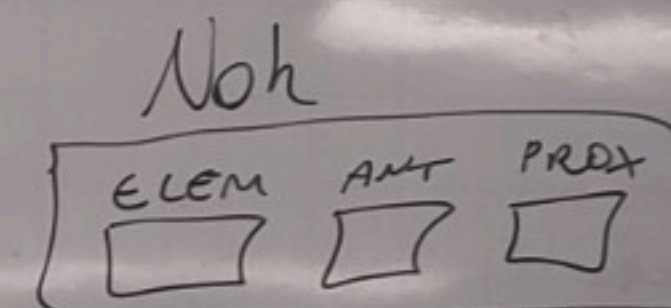
Noh *ESQ, *DIR;

//NULL

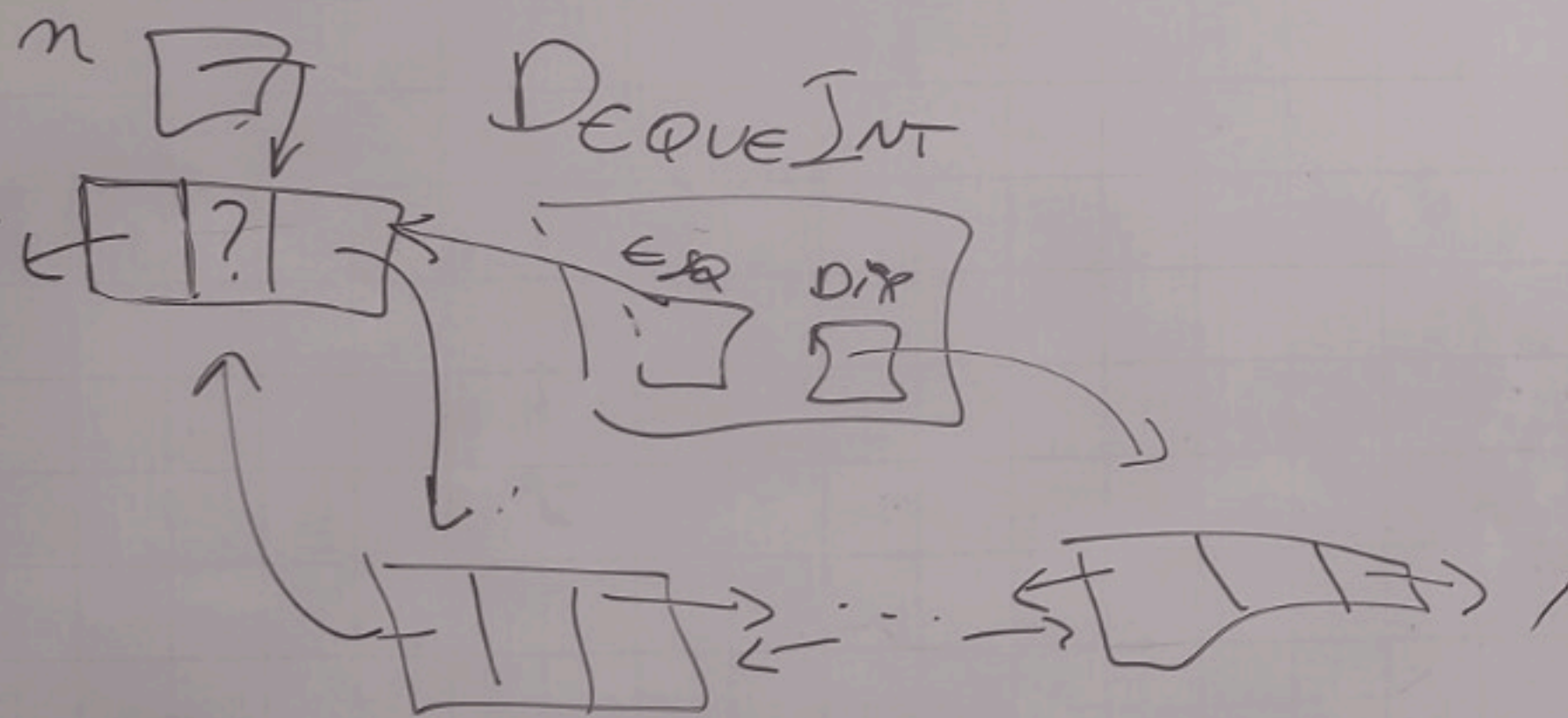
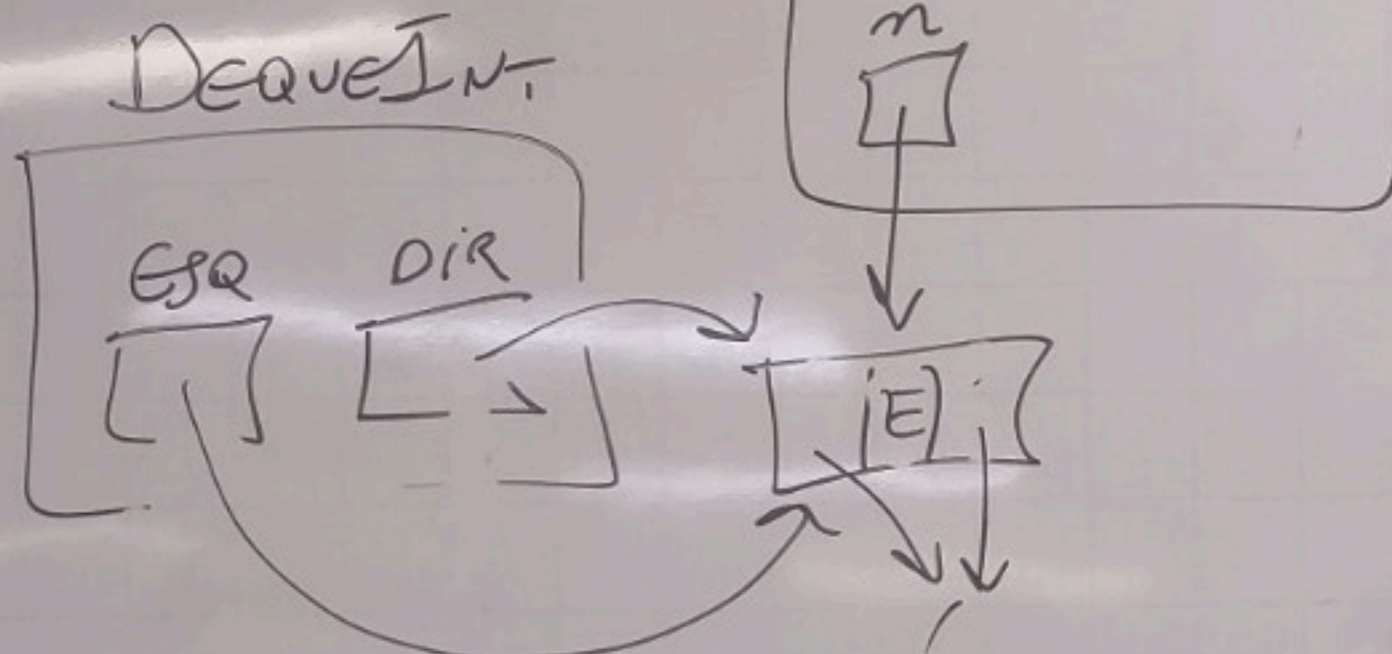
DequeInt() { ESQ = DIR = NULLPTR; }

VOID INSERIR_ESQ (INT E)

```
{
  Noh *n = NEW Noh; n->ELEM = E;
  if (ESQ == NULLPTR) // AND DIR == NULLPTR
  {
    ESQ = DIR = n;
    n->ANT = n->PROX = NULLPTR;
    n->ELEM = E;
  }
}
```



1. INSERIR-ESQ(E) 6



*: O 1º e o último
PODEM SER O MESMO.

EXCEÇÕES // STD::BAD_ALLOC

```
#include <exception>
using std::exception;
#include <iostream>
using std::cin; using std::cout;
int main () {
    DequeInt d;
    cout << "n: "; int n; cin >> n;
    try {
        int *v = new int[n];
        ... USAR v ...
        delete[] v;
    } catch (exception & e) {
        cout << "EXCEÇÃO: " << e.what() << '\n';
        return 1;
    }
}
```


ELSE

{

ESQ \rightarrow ANT = n;

n \rightarrow PROX = ESQ;

ESQ = n;

n \rightarrow ANT = NULLPTR;

n \rightarrow ELEM = E;

struct E { - };

E e {c=0};

LEON.
ED.
FEUPE