

UNIVERSIDADE FEDERAL DO CEARÁ

CK0109-2019.2 - TO2 - ESTRUTURAS DE DADOS

AULA 10 - 2019-09-02

Pilha via Vetor

1. Um exemplo de uso ("main.cpp"):

```
#include <iostream>
using std::cin; using std::cout;
#include "Pilha.hpp"
int main ()
{
    Pilha<double> p;
    if (p.inicializar()) return 1;
    ↪ SE VERDADEIRO, ERRO.
```

//main

for(;;)

```
{
    cout << "Número para empilhar"
    << " (negativo para terminar): ";
```

```
double num; cin >> num;
```

```
if (num < 0) break;
```

```
if (p.empilhar(num)) return 2;
    ↪ SE VERDADEIRO, ERRO.
```

```
cout << "Os números digitados, do último"
    << " para o primeiro, foram: ";
```

```
while (!p.vazia())
```

```
{ cout << ' ' << p.top();
```

```
  if (p.desempilhar()) return 3;
```

```
}
```

```
p.terminar();
```

```
} //main
```


2. UMA IMPLEMENTAÇÃO DE PILHA ('PILHA.HPP')

```
// OMITINDO GUARDAS DE INCLUSÃO...  
#include <new>  
using std::nothrow;  
  
template <typename T>  
struct Pilha  
{  
    T *v; int ult, tam_v;  
  
    bool inicializar()  
    {  
        v = new(nothrow) T[1];  
        if (v == nullptr) return true;  
        ult = -1; tam_v = 1;  
        return false;  
    }  
};
```

// PILHA

```
void TERMINAR()  
{  
    delete[] v;  
}  
  
bool VAZIA()  
{  
    return (ult == -1);  
}  
  
// REDIMENSIONAR fica AQUI.  
bool EMPILHAR(T e)  
{  
    if (ult == tam_v - 1)  
    {  
        if (REDIMENSIONAR(tam_v * 2))  
        {  
            return true;  
        }  
    }  
    ++ult; v[ult] = e;  
    return false;  
}
```


// PILHA

T topo() { return v[ULT]; }

// PRÉ-COND.: PILHA NÃO VAZIA.

bool desempilhar() // PRÉ-COND.: Ñ VAZIA

{

--ULT;

if(!vazia() &&

4 * (ULT + 1) <= TAM - v)

{

if(redimensionar(TAM - v / 2))

{ return true; }

}

return false;

}

// PILHA // PRÉ-COND.: $ULT + 1 \leq TAM - w$

bool redimensionar(int TAM - w)

{

T * w = new (nothrow) T[TAM - w];

if(w == nullptr) return true;

for(int i = 0; i <= ULT; ++i)

{ w[i] = v[i]; }

delete[] v; v = w;

TAM - v = TAM - w;

return false;

}

} // PILHA

// GUARDAS DE INCLUSÃO...