



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Estrutura de Dados

Aula 5

Estrutura: PILHAS

Representadas por Contiguidade Física
Projeto PilhaCF



Estruturas de Dados Lineares

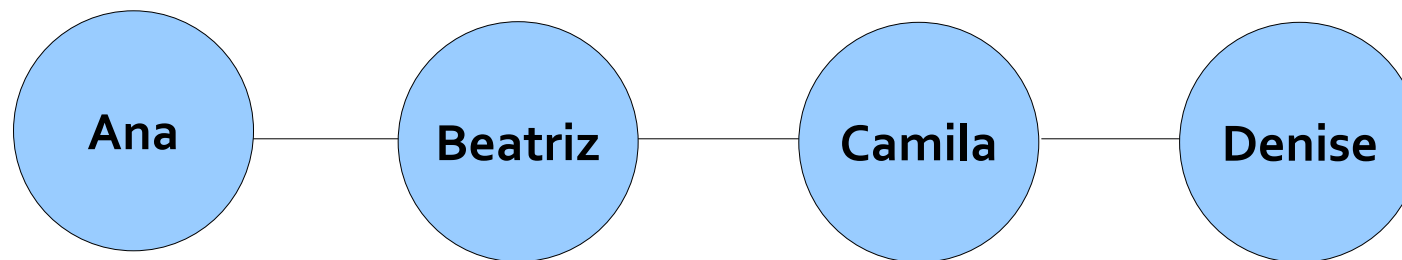
Como podemos representar estruturas lineares?

Estruturas de Dados

Estudo das técnicas de representação e manipulação de dados.

Problema: Obter uma relação dos alunos matriculados na disciplina de Estruturas de dados classificada em ordem alfabética.

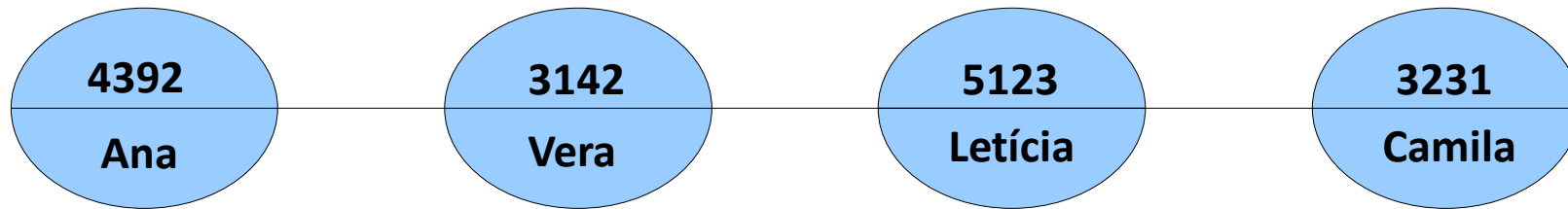
Estrutura: **Lista Linear**



Como representar na memória?

Operações? **inserir, remover, percorrer, procurar**

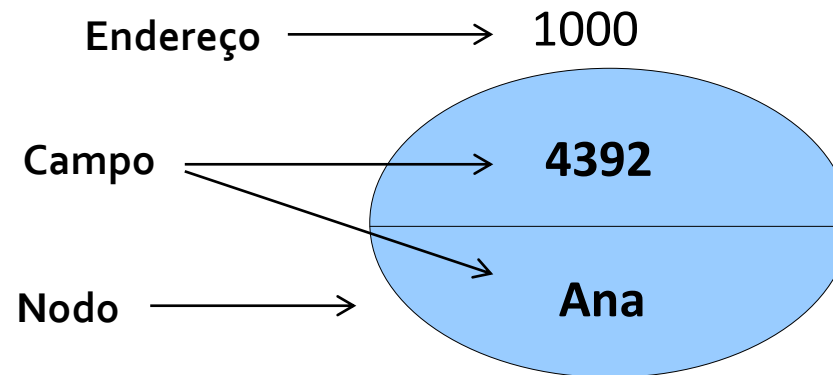
Lista Linear



Nodo ou nó: É a unidade básica de informação de uma certa estrutura.

Campo de um nodo: É a subdivisão de um nodo.

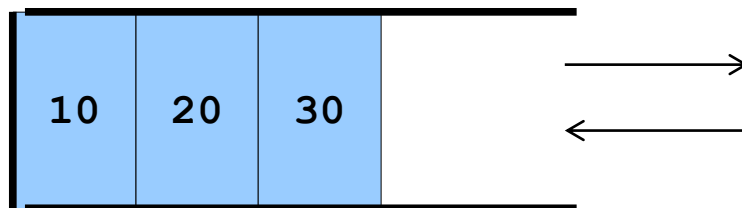
Endereço de um nodo: É a localização do nodo dentro de uma área de armazenamento.



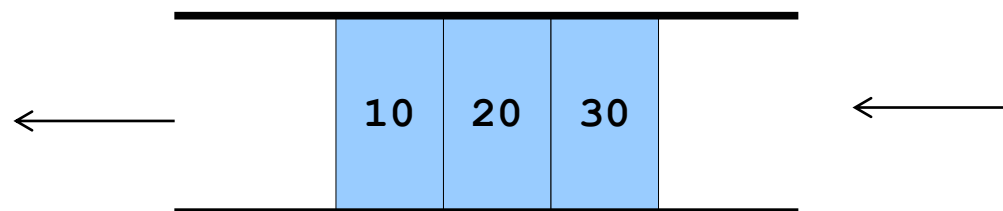
Listas Lineares com Disciplina de Acesso

São listas lineares onde o acesso a seus nodos é realizado apenas nas extremidades.

- Pilha (Stack)



- Fila (Queue)



Representação de Estruturas

A implementação de Estruturas de Dados pode ser realizada de duas maneiras:

1. Representação por **Contiguidade Física**:

- Os nodos são armazenados em endereços contíguos ou igualmente distanciados um do outro.
- Os relacionamentos são representados pela disposição física dos componentes na memória.
- A posição na estrutura lógica determina a posição na estrutura física.

Estrutura Lógica:

end 1	end 2	end 3
nodo 1	nodo 2	nodo 3

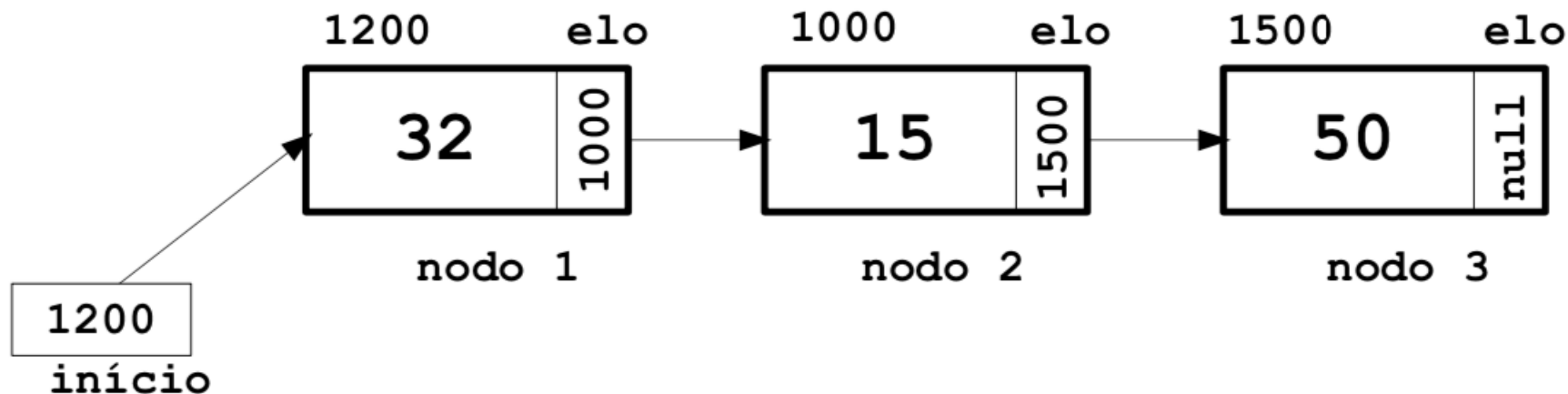
Estrutura Física:

Endereços		
1001	32	nodo 1
1002		
1003	15	nodo 2
1004		
1005	50	nodo 3
1006		
	Memória	

Representação de Estruturas

2. Representação por Encadeamento:

- A disposição física dos nodos independe de sua posição na estrutura lógica.
- Os relacionamentos são representados por elos que são ligações físicas explícitas.
- O valor contido em um campo de elo é o endereço de outro nodo.



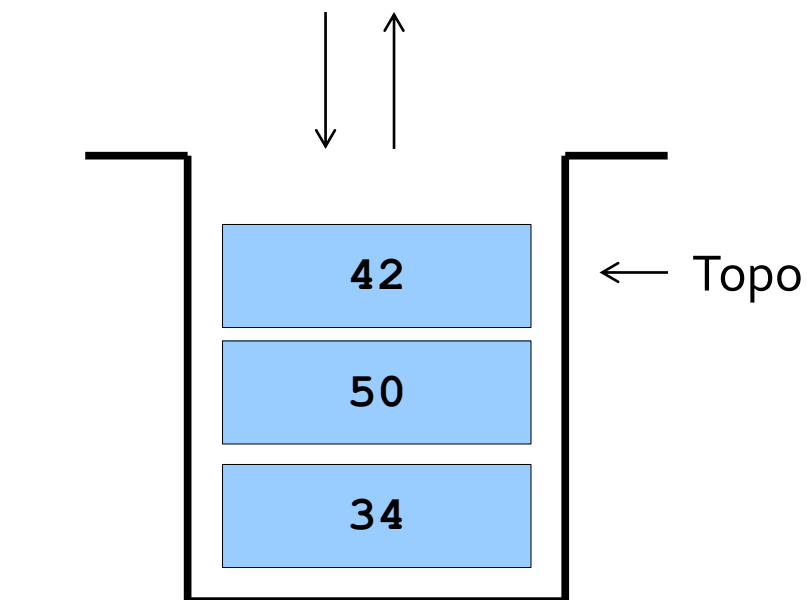


Trabalhando com PILHAS

Como modelar e criar operações com pilhas?

Pilha – representada por Contiguidade física de memória

É uma lista onde os acessos são realizados em uma única extremidade.



Critério "LIFO" (Last In First Out)
(último que entra é o primeiro que sai)

topo: Contém o **índice** do elemento que está no topo da lista

Estrutura Física:

Endereços		
1001	34	nodo 1
1002		
1003	50	nodo 2
1004		
1005	42	nodo 3
1006		
1007		nodo 4
1008		
1009		nodo 5
1010		
1011	2	topo
1012		

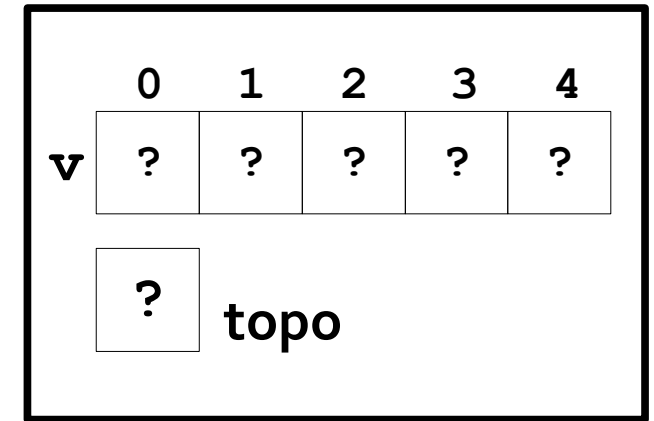
Pilha: por Contiguidade Física

```
// códigos de retorno
#define SUCESSO 0
#define PILHA_VAZIA 1
#define PILHA_CHEIA 2

// máximo de elementos
#define MAX_NODOS 5

typedef struct {
    int v[MAX_NODOS];
    int topo;
} PilhaCF;
```

Estrutura da Pilha:



v : Vetor que armazena os nodos da pilha

topo : índice do nodo que está no topo

Pilha: por Contiguidade Física

Operações:

```
void criaPilha(PilhaCF *p)
int  empilha (PilhaCF *p, int dado)      (push)
int  desempilha(PilhaCF *p,int *dado)    (pop)
int  estaCheia(PilhaCF p)
int  estaVazia(PilhaCF p)
int  consulta(PilhaCF p,int *dado)      //o dado do topo da pilha
```



PILHAS: Operação Empilha

Como modelar e criar operações com pilhas?

```
void criaPilha(PilhaCF *pl) ;
```

```
int main() {  
    PilhaCF pl;  
    criaPilha(&pl) ;  
    ...  
}
```

```
void criaPilha(PilhaCF *pl) {  
    pl->topo= -1;  
}
```

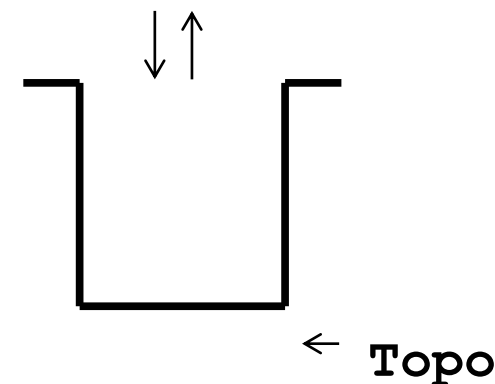
Inicializar uma Pilha (criaPilha)

Função: criaPilha

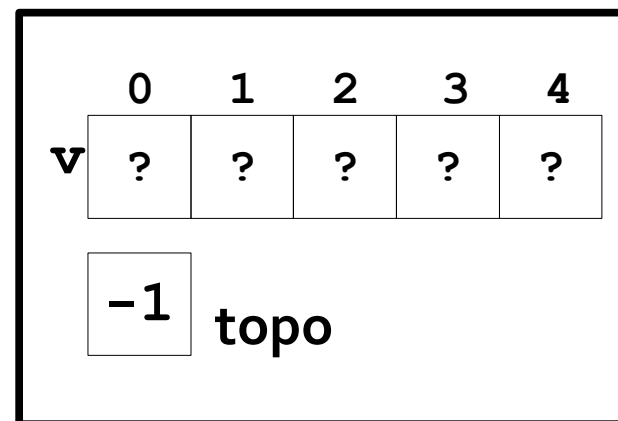
Saída: uma pilha vazia

Retorno: nenhum

Descrição: Cria uma pilha vazia



pilha vazia



```

int main(){
    ...
    int d;
    printf("Digite a idade: ");
    scanf("%d", &d);
    if (empilha(&pl, d)==SUCESSO)
        printf("Operação Realizada!")
    else
        printf("Pilha Cheia!");
    ...
}

```

Empilhar um dado (empilha)

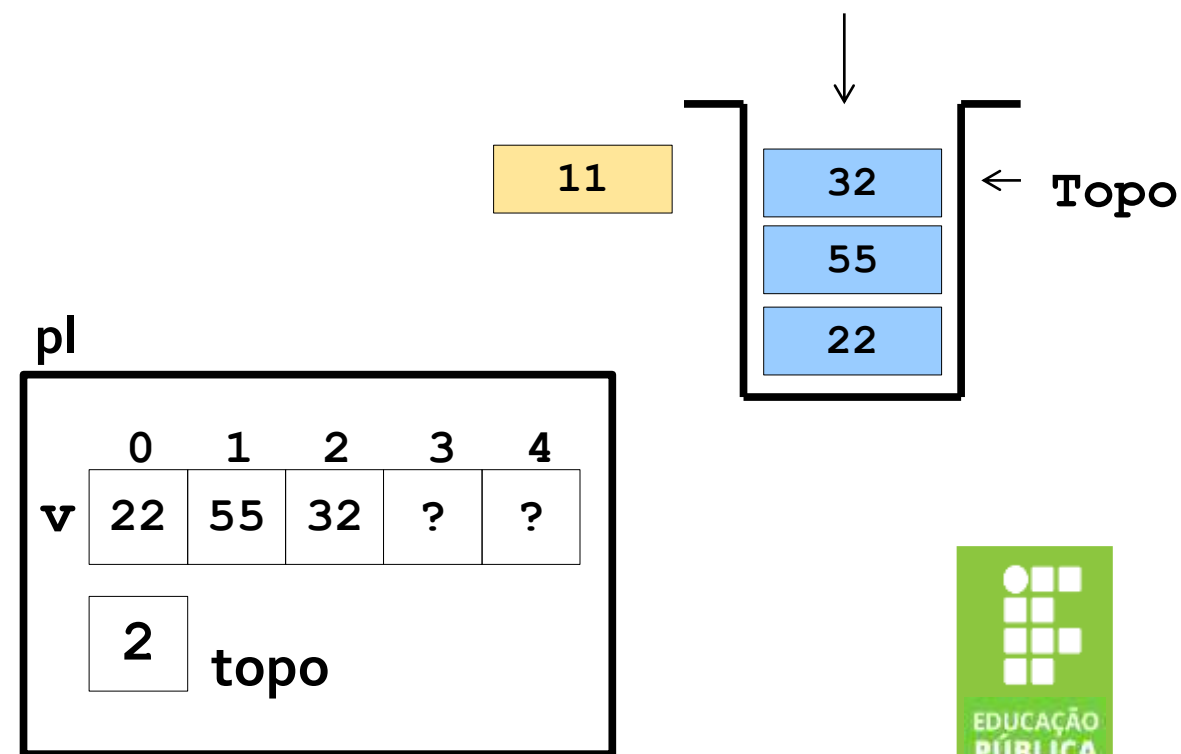
Função: empilha

E/S: uma pilha

Entrada: um dado do tipo **int**

Retorno: SUCESSO ou PILHA_CHEIA

Descrição: Empilha o dado na pilha



Empilhar um dado (empilha)

Função: empilha

E/S: uma pilha

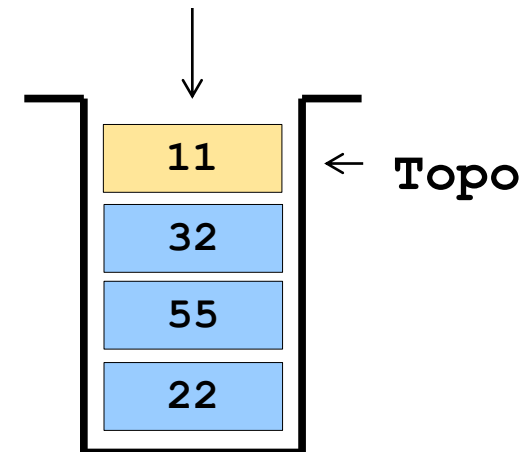
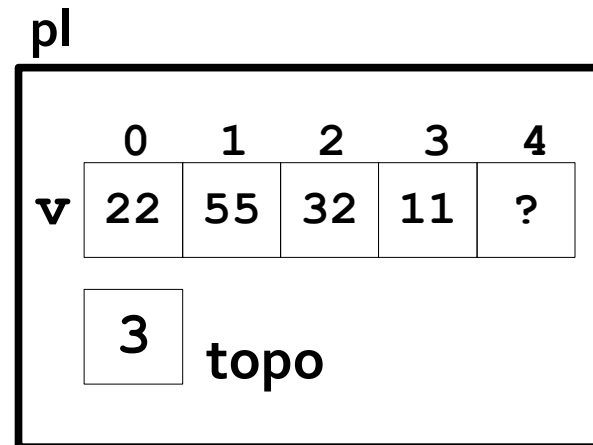
Entrada: um dado do tipo `int`

Retorno: `SUCESSO` ou `PILHA_CHEIA`

Descrição: Empilha o dado na pilha

```
int main(){
    ...
    int d;
    printf("Digite a idade: ");
    scanf("%d", &d);
    if (empilha(&pl, d)==SUCESSO)
        printf("Operação Realizada!")
    else
        printf("Pilha Cheia!");
    ...
}
```

```
int empilha (PilhaCF *pl, int d){
    if (pl->topo == MAX_NODOS-1)
        return(PILHA_CHEIA);
    else {
        pl->topo++;
        pl->v[pl->topo]= d;
        return(SUCESSO);
    }
}
```





PILHAS: Operação Desempilha

Como modelar e criar operações com pilhas?


```

int main() {
    ...
    int d;
    if (desempilha(&pl, &d)==SUCESSO)
        printf("Operação Realizada!")
    else
        printf("Pilha Vazia!");
    ...
}

```

Desempilhar um dado (desempilha)

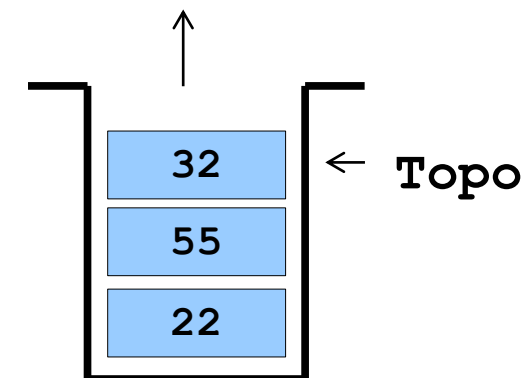
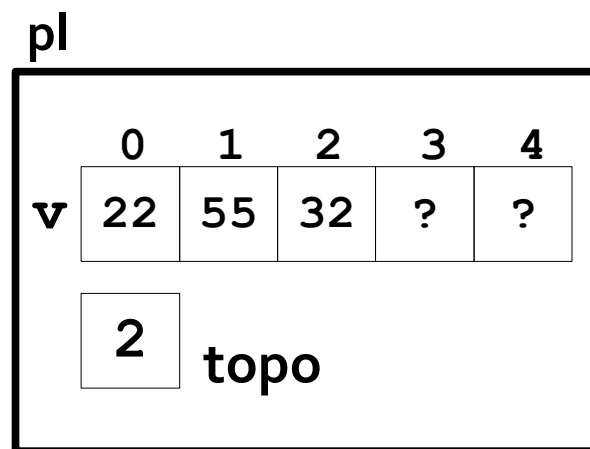
Função: desempilha

E/S: uma pilha

Saída: o dado desempilhado

Retorno: SUCESSO ou PILHA_VAZIA

Descrição: Desempilha um dado da pilha



Desempilhar um dado (desempilha)

```
int main() {  
    ...  
    int d;  
    if (desempilha(&pl, &d) == SUCESSO)  
        printf("Operação Realizada!")  
    else  
        printf("Pilha Vazia!");  
    ...  
}
```

```
int desempilha(PilhaCF *pl, int *d) {  
    if (pl->topo == -1)  
        return (PILHA_VAZIA);  
    else {  
        *d = pl->v[pl->topo];  
        pl->topo--;  
        return (SUCESSO);  
    }  
}
```

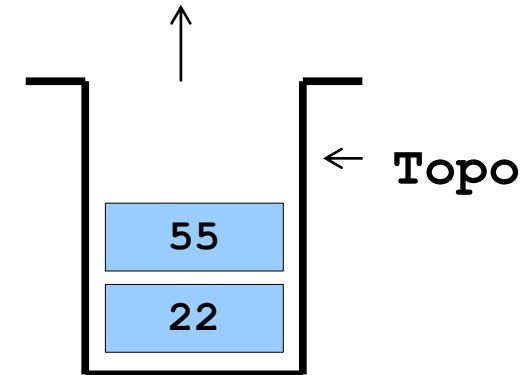
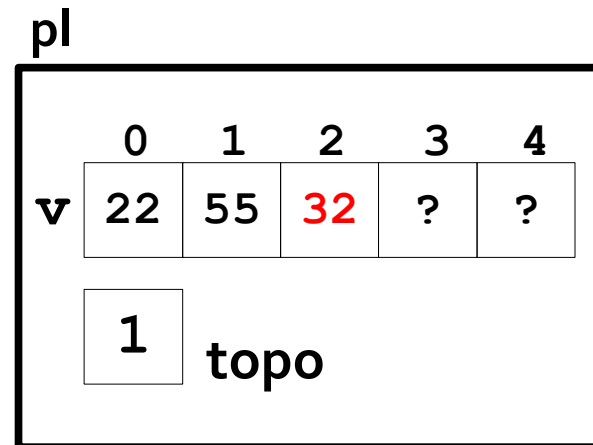
Função: desempilha

E/S: uma pilha

Saída: o dado desempilhado

Retorno: SUCESSO ou PILHA_VAZIA

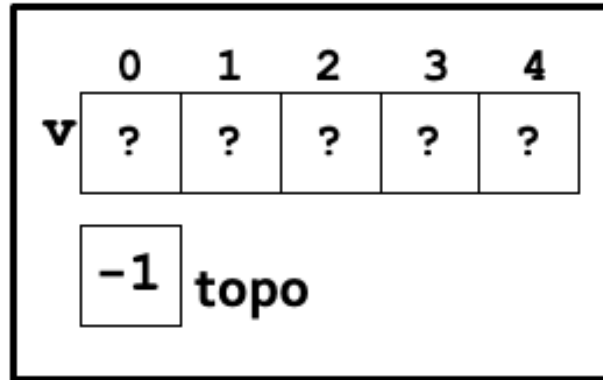
Descrição: Desempilha um dado da pilha



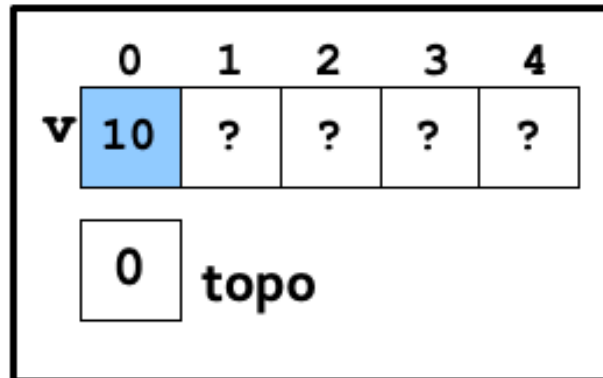
Pilha (Stack)

Exemplificando:

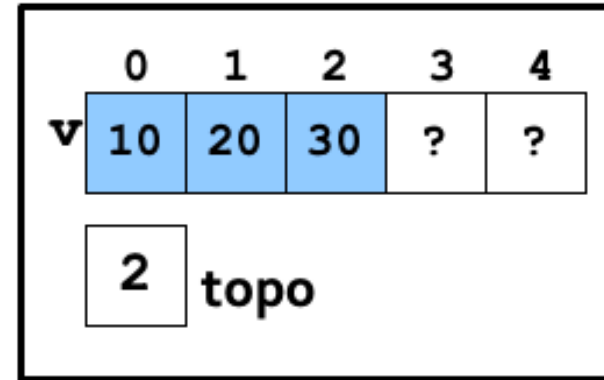
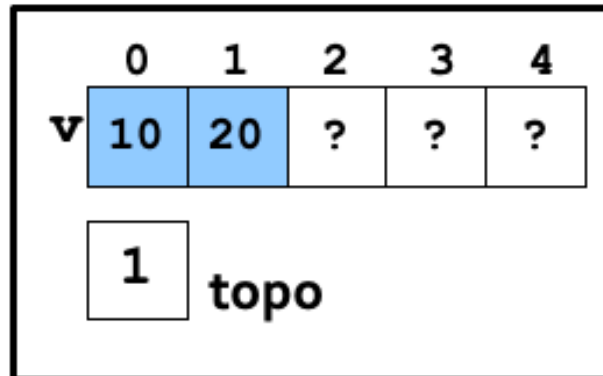
```
int main() {  
    PilhaCF pl;  
    criaPilha(&pl);
```



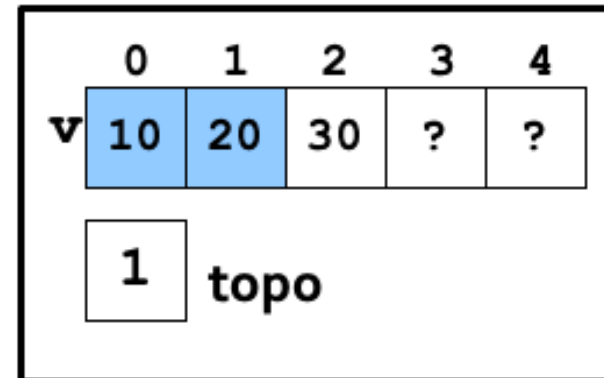
```
empilha(&pl,10);
```



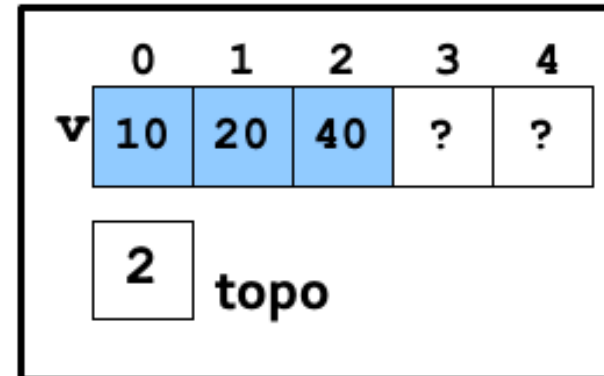
```
empilha(&pl,20);  
    ...
```



...
`empilha(&pl,30);`



`desempilha(&pl,&x);`



`empilha(&pl,40);`

Pilha: por Contiguidade Física

Operações:

```
void criaPilha(PilhaCF *p)
int  empilha (PilhaCF *p, int dado)      (push)
int  desempilha(PilhaCF *p,int *dado)    (pop)
int  estaCheia(PilhaCF p)
int  estaVazia(PilhaCF p)
int  consulta(PilhaCF p,int *dado)      //o dado do topo da pilha
```



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Estrutura de Dados

Aula 5

Estrutura: PILHAS

Representadas por Contiguidade Física
Projeto PilhaCF