The background of the slide features a large, faint, light green logo of UFES (Universidade Federal do Espírito Santo). The logo consists of a stylized, abstract shape resembling a flame or a sail, with the letters 'UFES' written in a bold, sans-serif font below it.

Estruturas de Dados I

(Listas lineares – Pilhas - Filas)

Prof. Jacson Luiz Matte
jacsonmatte@gmail.com

Estrutura de Dados

- É um conjunto de elementos agrupados sob um nome comum;
 - ▶ cada membro da estrutura pode ser constituído de um **tipo de dados diferente**;
- É uma forma de armazenar e organizar os **dados** em um computador;
- A implementação de uma estrutura de dados implica a escrita de procedimentos que criam e manipulam instâncias da estrutura;
- As estruturas e seus procedimentos facilitam a manipulação destes agrupamentos de dados;
 - ▶ Ex. uma ordenação pode ser feita como se todos os dados da estrutura fossem uma entidade única.

...Estrutura de Dados

- Por exemplo: podemos criar uma estrutura de dados para uma agenda de contatos;
 - ▶ como as informações serão armazenadas?
 - manter os contatos em ordem alfabética facilita a busca!
 - ▶ quais operações serão realizadas com a agenda?
 - criar, inserir, recuperar, remover, ordenar, atualizar contatos...
- A organização interna não precisa ser exposta ao usuário, mas somente uma **interface**.

...Estrutura de Dados

- Um outro exemplo: uma estrutura para armazenar informações sobre alunos;
 - ▶ nome, data de nascimento, data de ingresso, CPF, etc...
 - ▶ todos os campos estariam armazenados sob um mesmo nome (ex. aluno);
- Uma estrutura pode ser constituída de outras estruturas;
 - ▶ Ex.: estrutura aluno constituída por uma estrutura pessoa;

...Estrutura de Dados

- Definição de uma estrutura.

```
typedef struct Alunos{  
    char nome[40];  
    int matricula;  
    int ano_entrada;  
    char curso[20];  
    struct Data data_nascimento;  
}TpAluno;
```

```
struct Data{  
    int dia, mes, ano;  
};
```

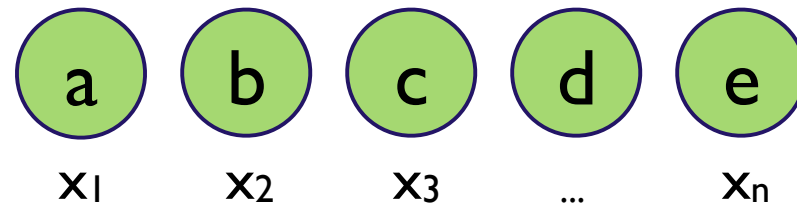
...Estrutura de Dados

- Como definir uma estrutura do tipo ALUNO?
 - ▶ `TpAluno fulano, beltrano, ciclano;`
- Esta declaração aloca espaço para três alunos, equivale a:
 - ▶ `int a, b, c;`
- Como referenciar um elemento(campo) da estrutura?
 - ▶ `fulano.ano_entrada = 2012;`
 - Quando `TpAluno` for um ponteiro:
 - `fulano->ano_entrada = 2012;`

Listas lineares

- Conjunto de elementos de mesmo tipo (nodos) entre os quais existe uma relação linear;
- Estruturas de dados flexíveis (podem crescer e diminuir de tamanho durante a execução de um programa);
- Os nodos da lista podem conter um dado primitivo ou um dado composto;

Listas Lineares



$n = 0$, lista vazia

$n > 0$, x_1 é o primeiro nodo e x_n é o último nodo

$1 < k < n$, x_k é precedido por x_{k-1} (para $k = 2, 3, \dots, n$) e sucedido por x_{k+1} (para $k = 1, 2, \dots, n - 1$.)

Operações sobre listas

- Para criar um TAD (Tipo Abstrato de Dados) lista, é necessário definir algumas operações que podem ser realizadas sobre ela.
- Operações necessárias à maioria das aplicações são:
 - Criar uma lista;
 - Inserir um novo nodo;
 - Retirar um nodo;
 - Localizar um nodo;
 - Ordenar os nodos da lista;
 - Excluir uma lista;
 - Combinar listas.

Pilhas

- Uma pilha é uma lista linear com a propriedade de aceitar inclusão e remoção somente no “topo” da lista;
- Dados são inseridos e retirados usando a estratégia LIFO (Last In, First Out);
- Podem ser implementadas com vetores (contiguidade física) e com listas encadeadas.
- Aplicações:
 - Pilha de execução de programas;
 - Avaliação de expressões.

Exemplo

- Algoritmo para avaliação de expressões pós-fixas:
 - ▶ Empilhar operandos até encontrar operador;
 - ▶ Retirar dois operandos; calcula e empilha o resultado;
 - ▶ Repetir até o final da expressão;

- A B / D E * + A -

A | B | /

A/B | D | E | *

A/B | D * E | +

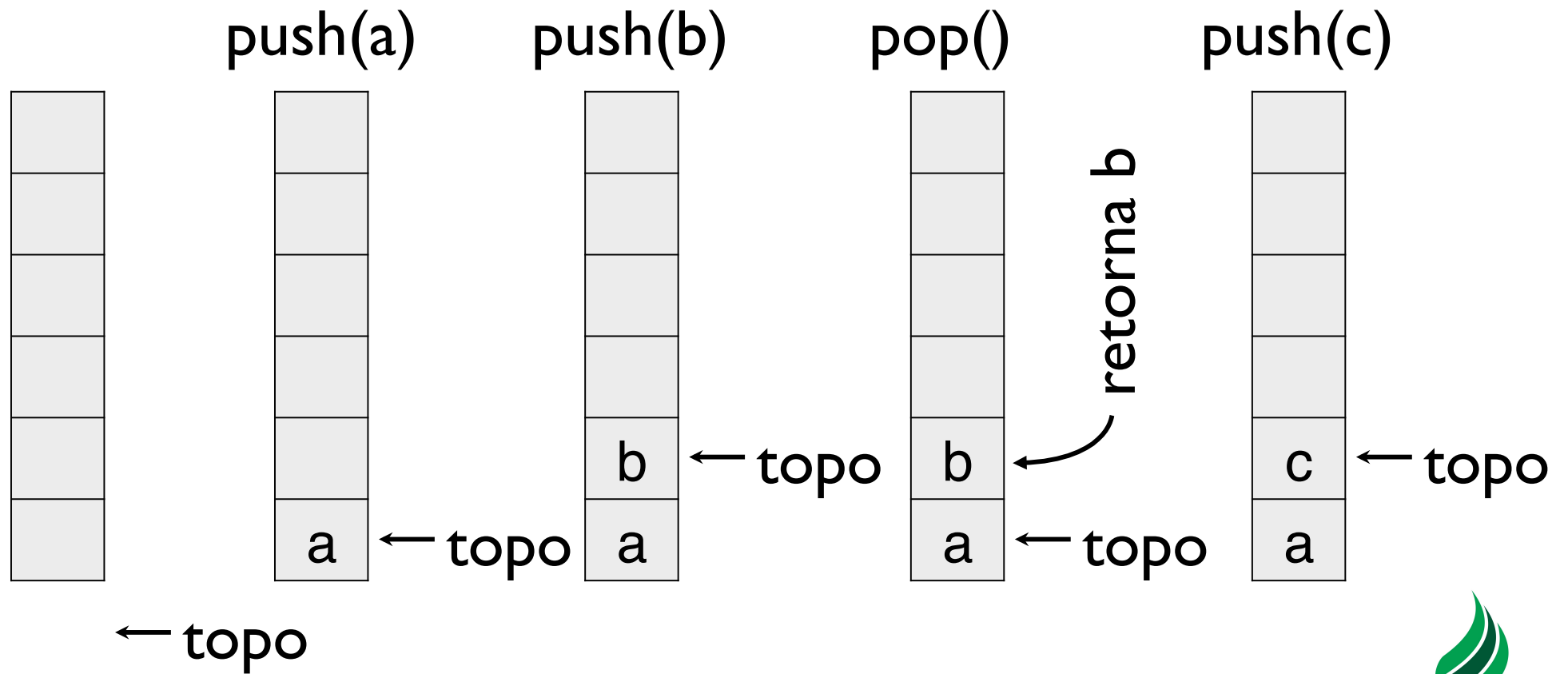
(A/B) + (D * E) | A | -

((A/B) + (D * E)) - A |

Pilhas

- Podemos implementar as seguintes operações:
 - ▶ Criar uma pilha;
 - ▶ Verificar seu estado (cheia ou vazia, não cheia, não vazia);
 - ▶ Inicializar a pilha;
 - ▶ Inserir um elemento na pilha (PUSH);
 - ▶ Extrair um elemento da pilha (POP);
 - ▶ Consultar/modificar o que está no topo da pilha;
 - ▶ Destruir a pilha, liberando as posições reservadas.

Funcionamento



Pilhas

- Implementação com vetores:
 - ▶ Podemos implementar uma pilha de no máximo n elementos com um vetor $p[0, \dots, n-1]$;
 - A pilha tem um atributo `topo` que indexa o elemento mais recente;
 - A pilha contém os elementos $p[0, \dots, topo]$, em que $p[0]$ é o elemento na parte inferior da pilha e $p[topo]$ o elemento no topo;
 - Quando `topo = -1`, a pilha está vazia.
 - Quando `topo = MAX`, a pilha está cheia.

```
typedef struct _pilha{  
    int vlr[MAX];  
    int topo;  
} pilha;
```

Pilhas - Exercícios

- Escreva um programa que use uma pilha para inverter a ordem das letras de cada palavra de uma string, preservando a ordem das palavras.
- Por exemplo, dado o texto `ESTE EXERCICIO E MUITO FACIL` a saída deve ser `ETSE OICICREXE E OTIUM LICAF`. (Lembre-se: strings em C terminam com `'\0'`).

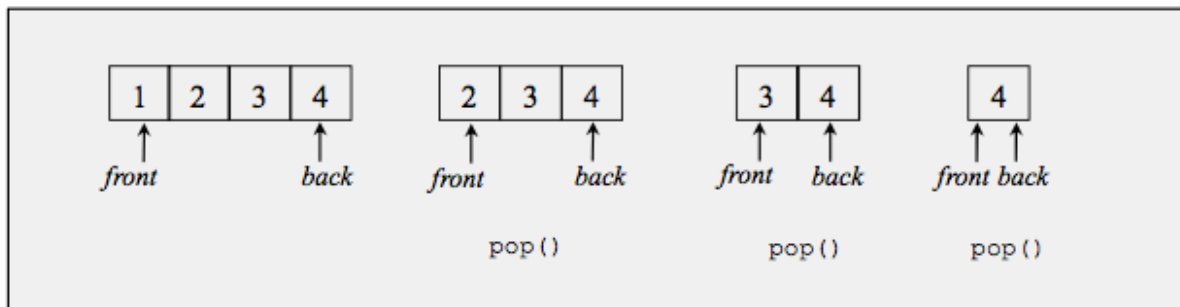
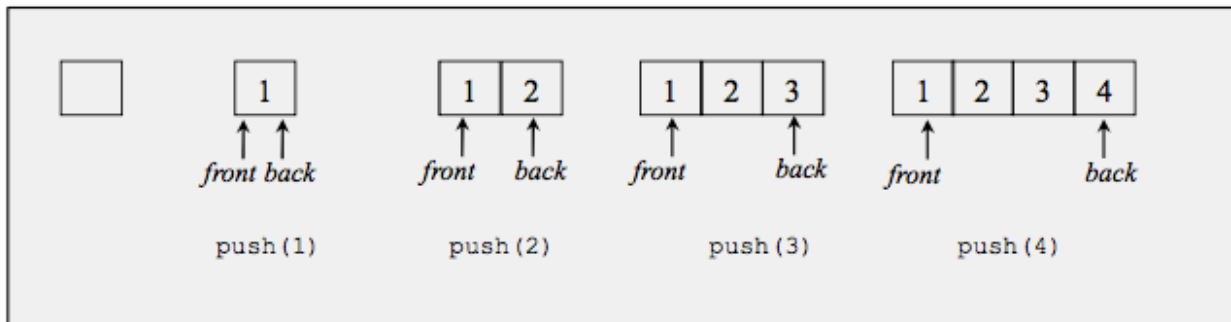
Filas

- Estrutura de dados linear em que:
 - Inserção de elementos se faz por uma extremidade chamada de **cauda**.
 - Remoção de elementos se faz por extremidade oposta a cauda, chamada de **cabeça**
 - Uma fila pode ser considerada uma restrição de lista
 - Estrutura do tipo FIFO (First-in-First-Out)
 - Exemplo:
 - Fila de atendimento em um supermercado



Filas

- Operações mais usadas:
 - Criar uma fila vazia
 - Adicionar ou remover um elemento da fila
 - Verificar qual elemento da cabeça/cauda da fila (mais antigo/recente)



Trabalho

- Implemente as seguintes funções em uma fila:
 - Inicializa fila
 - Inserção
 - Remoção
 - Verifica se está cheia
 - Verifica se está vazia

A fila deve conter a estrutura abaixo:

```
struct Node{  
    int num;  
    struct Node *prox;  
};  
typedef struct Node node;
```