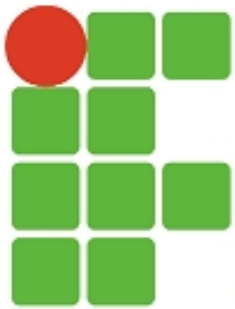


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA
Campus Campina Grande

5

Curso Engenharia da Computação

Programação e Estruturas de Dados

Pilhas – Fundamentos e Implementações

Prof. César Rocha
cesarocha@ifpb.edu.br

Objetivos

- *Explorar os conceitos fundamentais acerca do uso de **pilhas** utilizando a linguagem C*
  - *Organização e implementação, características, vantagens, desvantagens, regras de utilização, operações básicas e algoritmos de implementação*
- *Neste módulo, serão abordadas ainda as seguintes implementações: **seqüencial** e **encadeada***
- *Este módulo será utilizado como referência na entrega dos futuros projetos*
  - *Implementação das estruturas e algoritmos, criação das **bibliotecas** e práticas de laboratório*

Motivação

- *Uma das estruturas de dados mais simples é a* ***pilha***
  - *Possivelmente, por esta razão, a pilha é uma das estrutura de dados mais utilizada em programação*
- *A pilha vem sendo, inclusive, muito implementada pelo hardware da maioria das máquinas modernas*
  - *Lembre-se da* ***"pilha de chamadas"*** *de funções*
    - *Variáveis locais são empilhadas na pilha*
    - *Ao término da função, as variáveis são desempilhadas*
- *É um TAD onde as operações de inserção e retirada são efetuadas apenas no final da estrutura*

Motivação

- *A idéia fundamental da pilha é que todo o acesso a seus elementos é feito através do seu* ***topo***
  - *Seja na inserção ou na remoção de elementos*
  - *Assim, quando um novo elemento é introduzido na pilha, passa a ser o elemento do topo - e o único elemento que pode ser removido da pilha é o topo*
- *Isto faz com que os elementos da pilha sejam retirados na ordem inversa à ordem em que foram introduzidos inicialmente*
  - *O primeiro que sai é o último que entrou*
  - *Estruturas tipo LIFO (**last in, first out** strategy)*

Estratégia

- Podemos fazer uma analogia com uma pilha de pratos em um restaurante
  - Se quisermos adicionar um prato na pilha, o colocamos no topo da pilha de pratos
  - Para pegar um prato da pilha, retiramos o do topo
- Ou ainda: um software que implemente a função UNDO (o famigerado "Crtl+Z")
  - A última ação feita será a primeira a ser desfeita
  - Não podemos retirar qualquer ação! Segue-se a ordem!
  - Não podemos inserir uma ação na pilha de ações caso ela não se torne a última!

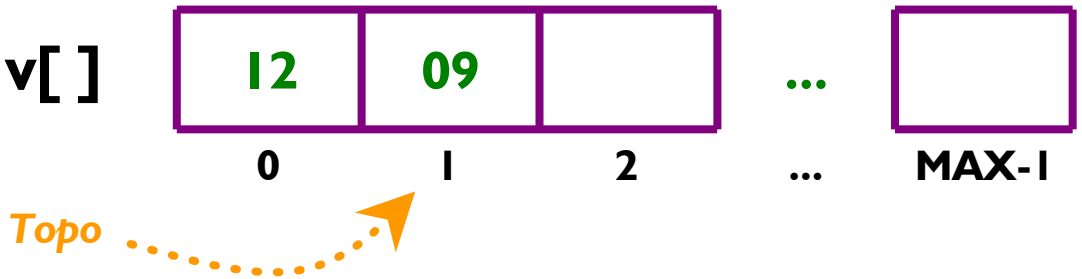
Propriedades

- Propriedades de uma pilha:
  - Existem **n** elementos empilhados;
  - **E1** é o elemento base da pilha;
  - **En** é o elemento topo da pilha;
- A inserção de um primeiro elemento **E1** em uma pilha vazia, torna-o o último a sair da estrutura
- Não se pode consultar qualquer elemento
- A inserção é sempre feita acima do elemento **En**
- A retirada é sempre feita no elemento **En**

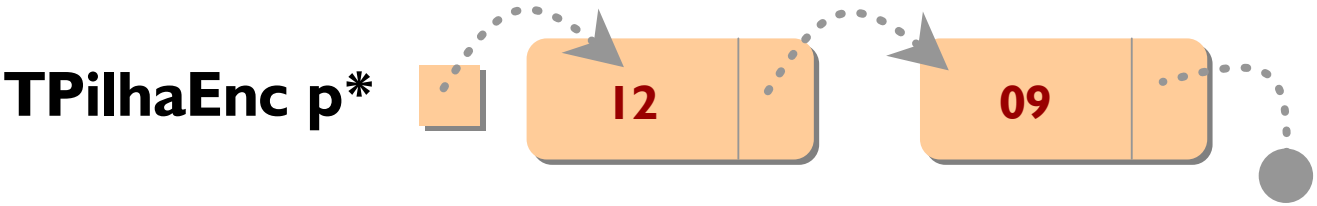
Tipos de pilhas

- *Neste estágio, estaremos trabalhando com dois tipos de pilhas:* ***seqüenciais*** *e* ***encadeadas***

  ❶ ***Seqüencial:*** *neste TAD, os elementos desta pilha são armazenados em endereços seqüenciais. Materializada na forma de um* ***vetor*** *(arranjo ou matriz).*



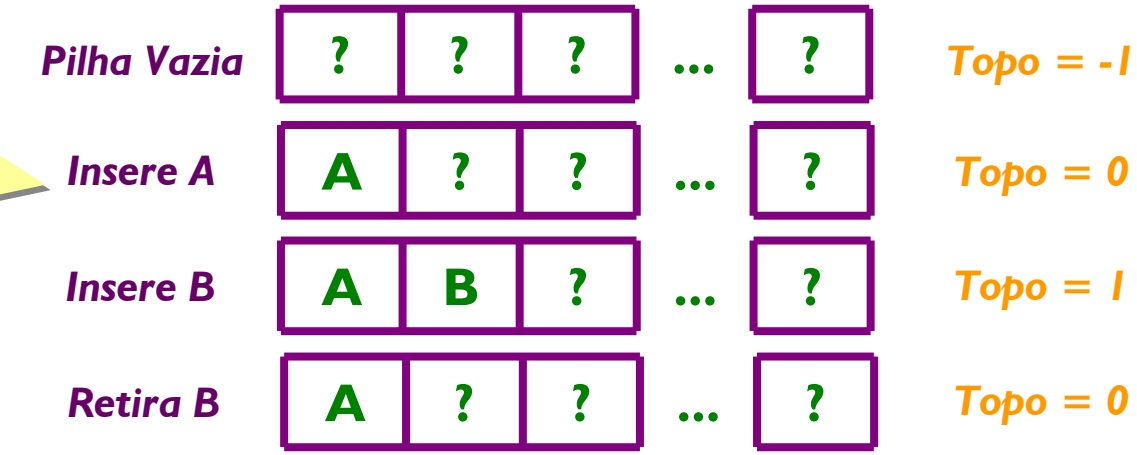
  ❷ ***Encadeada:*** *elementos encadeados por* ***ponteiros***



Pense um pouco...

- *O que você acha que seria necessário para implementar uma biblioteca de um novo TAD que representasse uma pilha seqüencial?*
  1. *um vetor de elementos (tamanho pré-definido)*
  2. *uma variável que controle o topo da pilha*

- ***É preciso deslocar os elementos a cada inserção ou remoção da pilha? Porquê?***
- ***Como poderíamos adaptar uma lista seqüencial a uma pilha?***



Pilhas seqüenciais – modus operandi

- *Iremos implementar este novo TAD utilizando as mesmas diretrizes das listas seqüenciais (já vistas)*
- *Teremos uma variável topo que irá servir de apoio para as inserções e remoções da pilha*
  - *E a variável* `posUltimo` *da lista seqüencial? Dá pra reaproveitar?*
- *Mas, as regras de inserção/remoção agora irão mudar!*
  - *Algumas funções da lista seqüencial (inserir ou remover mediante uma posição, entre outras) deverão ser eliminadas neste novo TAD*

Operações em pilhas seqüenciais

- A literatura é unânime quanto às operações básicas realizadas numa estrutura pilha seqüencial:
  1. **criar** uma pilha vazia
  2. verificar se uma pilha **está vazia**
  3. verificar se uma pilha **está cheia**
  4. **consultar o topo** da pilha (sem remover)
  5. **empilhar** um novo elemento
  6. **desempilhar** o elemento do topo
  7. **exibir** / imprimir os elementos de uma pilha

Pilhas encadeadas

- *O que fazer quando o número máximo de elementos na pilha não é conhecido?*
  - *Devemos implementar a pilha usando uma estrutura de dados dinâmica (com **alocação dinâmica**)*
  - *Podemos empregar os conceitos vistos nas listas simplesmente encadeadas*
- *Porém, se memória não constitui um problema na hora do armazenamento de dados, a pilha seqüencial (com vetores) pode parecer mais simples*
  - *Até porque não há mais os deslocamentos nas inserções e remoções de elementos na lista seqüencial*

Pilhas encadeadas

Pense mais um pouco...

- *O que você acha que seria necessário para implementar uma biblioteca de um novo TAD que representasse uma pilha encadeada?*
  1. *uma* `struct` *representando um nó da pilha (dado e ponteiro para o próximo nó da pilha).*
  2. *um ponteiro para o 1º nó representando a pilha*

- ***Como poderíamos adaptar uma lista encadeada a uma pilha?***
- ***Onde será o topo da pilha: no início ou no final da lista?***

Pilhas encadeadas - modus operandi

- *O primeiro elemento (início) da lista encadeada irá representar o topo atual da pilha*
  - *Cada novo elemento será inserido no início da lista e, conseqüentemente, sempre que solicitado, retiramos o elemento também a partir do início da lista*
- *Desta forma, vamos precisar de apenas duas funções auxiliares na lista:*
  1. *uma para inserir no início (empilhar)*
  2. *outra para remover do início (desempilhar)*
- *Ambas as funções retornam o novo primeiro nó da lista (o topo)*

Operações em pilhas encadeadas

- *A literatura é unânime quanto às operações básicas realizadas numa estrutura pilha encadeada:*
  1. ***criar** uma pilha vazia*
  2. *verificar se uma pilha **está vazia***
  3. *verificar se uma pilha **está cheia***
  4. ***consultar o topo** da pilha (sem remover)*
  5. ***empilhar** um novo elemento*
  6. ***desempilhar** o elemento do topo*
  7. ***exibir** / imprimir os elementos de uma pilha*

Algoritmos em C

- *O que deverá ser feito pelo aluno:*
  - *Escolha e instalação do ambiente a ser trabalhado no laboratório*
  - *Modelagem deste TAD (dados e operações)*
  - *Implementação dos algoritmos de operações básicas vistos em sala de aula na linguagem C*
  - *Utilização das regras de modelagem vistas no módulo anterior (criação de bibliotecas) e modularização*
  - *Implantação de código legível e bem documentado*
  - *Nomes de variáveis condizentes com o problema*
  - *Prática de laboratório*

Para um bom aproveitamento:

- *O aluno deve identificar a relação entre TAD (biblioteca e modularização) com a implementação da lista encadeada no código!*
- *Resolva todas as questões da* ***prática de laboratório de pilhas sequencial e encadeadas***
- *Procure o professor ou monitor da disciplina e questione conceitos, listas, etc.*
- *Não deixe para codificar tudo e acumular assunto para a primeira avaliação.*
  - *Este é apenas um dos assuntos abordados na prova!*