

A partir da implementação do TAD pilha (stack), é possível realizar atividades de maior complexidade, implementando aplicações úteis para software, algoritmos e outros serviços.

O objetivo deste trabalho é, a partir da implementação de uma pilha, criar uma simulação de um sistema de organização de URLs, na qual o usuário pode “ir para frente” ou “voltar” a página, assim como em navegadores web.



*Exemplo de sistema de navegação de páginas do Google Chrome*

A implementação pode ser realizada pela linha de comando, fornecendo um *prompt* ao usuário para que insira novas páginas web. Em seguida, o programa deve permitir que o mesmo navegue entre a página atual, as “páginas seguintes” e as “páginas anteriores”.

Caso o usuário insira uma nova página, a página atual deve ir para as “páginas anteriores” (<-) e ser substituída pela inserção. Caso o usuário retorne à página anterior, a página atual deve ser colocada nas “páginas seguintes” (->). O usuário deve ser capaz de alternar entre voltar a página, avançar a página ou inserir novas URLs. Os comandos requeridos pelo trabalho são:

- **novo**, para uma nova URL.
- **anterior**, para retornar uma URL.
- **posterior**, para avançar uma URL.
- **sair**, para encerrar o processo.

Abaixo estão alguns exemplos do funcionamento esperado do programa:

```
> novo google.com
Página atual: google.com

> novo wikipedia.org
Página atual: wikipedia.org

> anterior
Página atual: google.com
```

*Inserção de novas páginas e retorno para uma página anterior:*

```
> posterior  
Página atual: wikipedia.org
```

*Avanço para uma página posterior:*

```
> novo google.com  
Página atual: google.com  
  
> novo wikipedia.org  
Página atual: wikipedia.org  
  
> anterior  
Página atual: google.com  
  
> novo youtube.com  
Página atual: youtube.com  
  
> anterior  
Página atual: google.com  
  
> posterior  
Página atual: youtube.com
```

*Fluxo de avanço e retorno entre páginas. Nota-se que a página wikipedia.org se perdeu após o retorno e a inserção de uma nova página.*

Como mostrado nos exemplos, inserir uma nova página inicia um novo fluxo de navegação, então as “páginas seguintes” devem ser limpas. Por exemplo, no fluxo seguinte:

- **novo** youtube.com
- **novo** google.com
- **anterior**
- **novo** wikipedia.org
- **posterior**

O último comando deve resultar em um erro, já que o usuário retornou e depois iniciou um novo fluxo de navegação, perdendo assim acesso à URL anterior “youtube.com”.

## Submissão

O trabalho será realizado em duplas. Submeta a implementação da dupla (apenas um membro precisa submeter) na plataforma TIDIA-AE na Atividade “Trabalho 1 - Stack” até a data limite lá especificada. Coloque seus nomes completos em todos os arquivos sendo submetidos, na forma de comentário no início de cada arquivo (.h ou .cpp). Compacte os arquivos em um único arquivo .zip (não utilize espaços no nome do arquivo compactado, nem adicione pastas/diretórios no arquivo compactado):

## Avaliação

Na nota do trabalho também serão considerados os seguintes critérios (além dos já mencionados nas Disposições Gerais entregues no início do semestre):

**Correção:** O programa faz o que foi solicitado? Faz tudo o que foi solicitado? Utiliza encapsulamento de informação? (i.e., acessa adequadamente os ADTs definidos?) Não há vazamento de memória?

**Eficiência:** As operações são executadas da maneira mais eficiente para cada estrutura de dados? Evita código duplicado/redundante/não atingível?

**Interface:** É simples de usar, genérico, prático, tolera os erros mais óbvios? O trabalho foi entregue dentro das especificações, implementando um ADT? Os arquivos estão em formato ZIP, com os nomes de arquivos solicitados?

- o interface do programa;
- o implementação dos ADTs e métodos;

**Código fonte:** é claro (*layout*, espaçamento, organização em geral), nomes de variáveis são sugestivos, e há documentação/comentários apropriados no código? Faz uso de pré- e pós-condições? Quando aplicável, faz uso de subalgoritmos (funções, procedimentos ou métodos) adicionais que melhoram a legibilidade do(s) método(s) solicitado(s) sem comprometer sua eficiência (por exemplo, na notação assintótica  $O(n)$ , onde  $n$  representa o tamanho da entrada?)

---