

# Gerenciando Abas no Navegador

## SCC0201 — Trabalho 5

Prof. Moacir A. Ponti + PAEs: Edesio, Leonardo + Monitores: João, Giuliano

26 de outubro de 2018

Nesse trabalho vamos implementar um programa que gerencie abas em um navegador utilizando listas encadeadas.

## 1 Tarefa

Analú adora abrir múltiplas abas em seu navegador. As vezes as abas são tantas que quase não dá pra ver mais o nome nem o ícone, mas ela quase nunca fecha uma aba, só vai abrindo outras. Ajude Analú implementando uma forma de gerenciar as abas.

## 2 Especificações

O programa deve simular a abertura de novas abas e operações sobre essas. Você deve usar uma lista encadeada para armazenar as abas, as quais possuem quatro informações: Título (com até 30 caracteres), URL (1024 caracteres), data (dia e mês) e hora (hora de 0 a 23 e minuto) da abertura da aba.

O programa deve permitir:

1. Abrir nova aba (incluir na lista encadeada), a qual sempre é inserida no final da lista;
2. Alterar a posição de uma aba (reposicionar na lista encadeada): considerando que as abas tem posições numeradas 1, 2, 3, ...;
3. Ordenar as abas com base na data e hora em que foram abertas.

A sugestão de solução é a implementação de algoritmos de ordenação radix sort que utilizam o bucket sort como subrotina.

### 2.1 Instruções

As instruções são formadas por um código de identificação e recebem ou não informações extra via teclado. As instruções possíveis são **abrir**, **alterar**, **ordenar**, **exibir** e **sair**.

1. **Abrir**: será acessada pela opção 1, e a seguir :
  - recebe as informações da aba: título, url, data e hora,
  - inclui a aba na lista encadeada.
2. **Alterar**: será acessada pela opção 2, e a seguir:
  - recebe a informação do título da aba a ser alterada,
  - recebe a nova posição (um número inteiro) da aba, que pode ser 1 ou maior,

- reposiciona o nó da aba de forma que fique na nova posição informada,
- caso haja mais do que um título igual, altere apenas o primeiro encontrado, percorrendo desde o início da lista de abas até o fim,
- se o título informado não existir, o programa não deve fazer nada e retornar ao menu de opções,
- caso a nova posição exceda o total de abas, posicionar como última aba.

3. **Ordenar:** será acessada pela opção 3, e a seguir:

- reordena as abas utilizando como chave: a data (na ordem mes e dia) seguido pela hora (na ordem: horas e minutos);
- utilizar um algoritmo para ordenação em tempo linear e implementado por você (não será aceita implementação pronta nem utilizando pacotes ou bibliotecas).

4. **Exibir:** listar na tela a configuração das abas, cada aba em uma linha no formato:

```
[titulo] [url] [dd/mm] [hh:mm]\n
...
[titulo] [url] [dd/mm] [hh:mm]\n\n
```

Note que a última linha impressa deve ter uma quebra de linha `\n` extra.

5. **Sair:** opção 5, desaloca toda a memória alocada, inclusive as abas, e finaliza o programa.

## 2.2 Exemplo

Abaixo, abrimos 3 abas, alteramos uma das abas para a posição 1, outra aba para a posição 10 (posição final), exibimos as abas, ordenamos, exibimos novamente e finalizamos o programa:

```
1
USP
http://www.usp.br
20 11
08 30
1
ICMC USP
http://www.icmc.usp.br
20 11
09 15
1
FOG
http://www.fog.icmc.usp.br
21 11
21 30
2
FOG
1
2
USP
3
4
3
4
5
```

Saída:

FOG <http://www.fog.icmc.usp.br> 21/11 21:30  
ICMC USP <http://www.icmc.usp.br> 20/11 09:15  
USP <http://www.usp.br> 20/11/2018 08:30

USP <http://www.usp.br> 20/11 08:30  
ICMC USP <http://www.icmc.usp.br> 20/11 09:15  
FOG <http://www.fog.icmc.usp.br> 21/11 21:30

### 3 Entrega e Avaliação

O trabalho será avaliado levando em consideração:

1. Uso de listas encadeadas para gerenciar a lista de abas: o não uso dessa estrutura de dados irá invalidar o trabalho entregue;
2. Realização dos objetivos / lógica utilizada;
3. Uso de comentários e estrutura no código (e.g. indentação,; legibilidade, modularização);
4. Resultado da plataforma run.codes;
5. Eficiência da implementação.