



**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO**

---

Disciplina: Estrutura de Dados I	Turma: CC3/EC4	Data: 22/09/14	Nota:
Professor: Renato E. N. de Moraes	Semestre: 2014-2	Valor: 2,0 pt	
Aluno:	Trabalho 01 - Ônibus		

Implemente algoritmos de lista encadeada e de lista duplamente encadeada cicular para a realização da seguinte aplicação: Sistema de Informação de Linhas de Ônibus Intermunicipal.

Imagine um terminal de informações em uma rodoviária onde um passageiro potencial pode se informar se existem linhas de ônibus que passam por este terminal ou que partem dele e que vão para a cidade para a qual o passageiro quer ir. O sistema funcionará da seguinte forma:

- O usuário chega no sistema e, defrontado com uma telinha de informações bonitinha, digita o destino desejado e o horário aproximado que deseja chegar no destino informado.
- Nesse sistema cada linha de ônibus possui um número que a identifica.
- O sistema devolverá, como resposta, o número da linha, a companhia e o horário de partida do ônibus do terminal e o horário de chegada do ônibus no destino.

O sistema a ser implementado deverá armazenar a informação utilizando TADs para dois tipos de listas encadeadas diferentes:

- Uma lista encadeada simples de todas as linhas de ônibus, organizada por número da linha de ônibus. Este número tem 5 dígitos e é único para o país inteiro.
  - Cada elemento desta lista possui dados sobre número da linha e a companhia e um ponteiro para uma lista com todas as paradas.
- Uma lista duplamente encadeada circular para cada linha de ônibus, onde cada *elemento* possui dados referentes a uma parada do ônibus.
  - Neste *elemento* estão informações sobre o nome da parada, o horário de chegada do ônibus na rodoviária e o horário de saída do ônibus da rodoviária.
  - Esta segunda lista é circular porque neste sistema não estão só as linhas de ônibus que iniciam nesta rodoviária, mas também linhas que vem de outros lugares. A lista também tem de representar o fato de que os ônibus vão e voltam e passam pelos mesmos pontos da rota duas vezes em horários diferentes.
  - Os *elementos* devem seguir a ordem dos horários.
  - Assim, a linha 23451 Florianópolis/Lages, que passa por Blumenau quando vai de Fpolis para Lages ficaria assim:

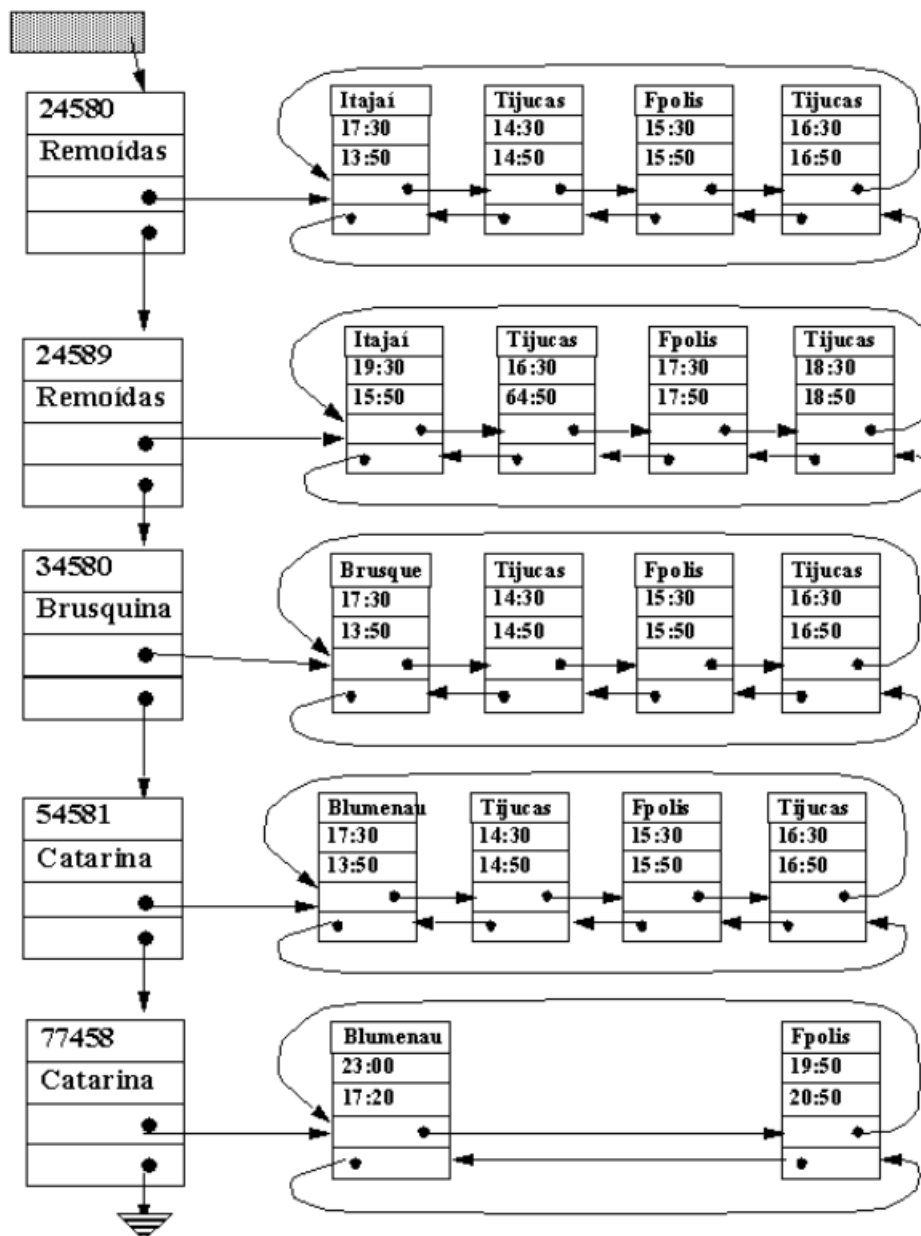
Fpolis -> Blumenau -> Rio do Sul -> Lages -> Bom Retiro -> Fpolis

sendo que o ônibus sai de Fpolis às 7:00 e retorna a Fpolis às 23:00.

- A lista de paradas deve ser duplamente encadeada porque o sistema vai procurar por um destino com um horário adequado em todas as listas circulares e, encontrando, percorrerá esta lista de trás para frente para encontrar o horário de saída.



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO



O programa deve ter as seguintes opções/comportamento:

- Um menu de usuário com as opções:
  - (1) Procura Linha de Ônibus
    - \* A procura da linha será feita como dito anteriormente, sendo que, quando um destino adequado é encontrado (adequado = tempo de chegada  $\geq$  tempo indicado pelo usuário), o programa percorre a lista onde encontrou o destino adequado de trás para frente até encontrar o terminal de origem, para informar ao usuário o horário de saída também.



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO

---

- (2) Manutenção do Programa. O usuário deve fornecer um arquivo cuja primeira informação é a identificação do funcionário que está realizando a manutenção (essa informação deve ficar armazenada nas alterações realizadas, exceto eliminações). As seguintes operações podem ser realizadas:
1. Incluir uma nova linha, onde o funcionário poderá incluir uma linha dando o número e o nome da companhia, sendo criado um nodo com um apontador nulo para as paradas da linha.
  2. Incluir uma parada numa linha. O usuário deve dizer qual o número da linha onde quer incluir uma parada. A posição em relação as demais paradas já existentes é definida pelo horário.
  3. Alterar o conteúdo de uma parada. A posição pode mudar se houver mudança nos horários da parada.
  4. Eliminar uma parada de uma linha.
  5. Eliminar uma linha. Elimina a linha dada e todas as paradas.

Você deve fazer testes de consistência se essas operações podem ser aplicadas e deve imprimir mensagens de sucesso ou falha (por exemplo, eliminar uma parada que não existe na linha). Além disso, procure escrever funções e procedimentos auxiliares que facilitem a implementação das operações acima, evitando a repetição desnecessária de código. Use TAD's. O seu programa deverá ler os dados de entrada a partir de um arquivo. Esse arquivo é basicamente uma lista de comandos (um por linha) em formato texto. Por exemplo:

CADASTRA <numero\_linha> <nome\_companhia> (ex. CADASTRA 24280 Remoidas).

O programa deve ser escrito em C, seguindo rigorosamente o cabeçalho de programa dado durante as aulas (usar o código dado nos laboratórios). Trabalho com codificação diferente terá nota zero.

O Trabalho é individual. O programa funcionando deve ser apresentado obrigatoriamente em uma máquina do laboratório no dia 21/10/2014. Os arquivos de entrada serão fornecidos pelo autor do trabalho no dia da apresentação. O professor poderá também fornecer arquivo próprio. Quando solicitado, o autor deverá explicar o código e justificar os resultados obtidos.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, TAD's, identificação e comentários no programa também vão valer pontos.
- Você NÃO deve utilizar estruturas de dados já implementadas na linguagem tais como `array_list`, `linked_list` ou similares. Ou seja, você deverá implementar todas as estruturas utilizadas.