



## Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

### Exercício sobre listas ligadas

**Objetivo:** Praticar a implementação de listas ligadas **com recursão**.

O endereço do sistema de submissão é o <https://les.dsc.ufcg.edu.br:8443/EasyLabCorrection>.

Relembre o conceito de lista simplesmente e duplamente ligada visto em sala de aula.

Atividades necessárias antes de iniciar o exercício:

1. Crie um projeto no Eclipse chamado LEDA, por exemplo (pode ser qualquer outro nome que lhe convier);
2. Descompacte o arquivo baixado (exceto o PDF) na pasta dos fontes (normalmente **src**) do seu projeto LEDA criado no seu workspace. O arquivo baixado tem a seguinte estrutura:
  - adt
  - linkedList
  - LinkedList.java (INTERFACE DA LISTA LIGADA)
  - DoubleLinkedList.java (INTERFACE DA LISTA DUPLAMENTE LIGADA)
  - RecursiveSingleLinkedListImpl.java (IMPLEMENTAÇÃO PARCIAL DA LISTA SIMPLES)
  - RecursiveDoubleLinkedListImpl.java (IMPLEMENTAÇÃO PARCIAL DA LISTA DUPLA)
  - stack
  - Stack.java
  - StackOverflowException.java.
  - StackUnderflowException.java.
  - StackRecursiveDoubleLinkedListImpl.java (IMPLEMENTAÇÃO PARCIAL DE UMA PILHA USANDO UMA LISTA DUPLA RECURSIVA)
3. No Eclipse, selecione a pasta dos fontes no projeto LEDA e faça um refresh (apertar F5). Note que deve aparecer os arquivos mencionados acima.

Agora você está pronto para começar a trabalhar nas seguintes atividades:

1. Observe a interface LinkedList.java. Ela descreve os serviços de uma lista genérica.
2. Observe a interface DoubleLinkedList.java. Ela descreve os serviços de uma lista duplamente encadeada genérica.
3. Observe também a existência implementação incompleta RecursiveSingleLinkedListImpl. Você precisa implementar os métodos incompletos (*com recursão*).
4. Observe também a existência implementação incompleta RecursiveDoubleLinkedListImpl. Você precisa implementar os métodos incompletos (*com recursão*). Note a herança existente entre as classes.
5. Observe também a existência implementação incompleta StackRecursiveDoubleLinkedListImpl. Ela representa uma pilha implementada usando uma lista dupla (recursiva) internamente. Você precisa implementar os métodos incompletos. Essa implementação é semelhante à sua implementação de pilha usando lista dupla iterativa.

6. Concentre-se em implementar conforme descrito na interface e pense em cenários para testar suas implementações de lista simples e dupla antes de enviar. Alguns cenários interessantes são: testes em uma lista vazia e em uma lista não vazia, nas inserções e remoções verificar se o tamanho da lista é alterado corretamente, após operações de inserção e remoção verificar se a estrutura interna (envolvendo apontadores) está correta, etc.

### Instruções para o envio

Ao terminar o exercício, faça os seguintes passos:

1. Compacte a pasta **adt** que existe nos fontes de seu projeto LEDA (**src**) e retire a classe TestLists.java desse arquivo compactado. A compactação DEVE ser feita a partir do diretório raiz de seus fontes de forma a preservar a estrutura de pastas que refletem a estrutura dos pacotes (package) Java. Por exemplo, você deve ter um arquivo compactado NOME\_COMPLETO\_DO\_ALUNO.ZIP com a seguinte estrutura:
  - adt
  - linkedList
  - RecursiveSingleLinkedListImpl.java ([IMPLEMENTAÇÃO PARCIAL DA LISTA SIMPLES](#))
  - RecursiveDoubleLinkedListImpl.java ([IMPLEMENTAÇÃO PARCIAL DA LISTA DUPLA](#))
  - stack
  - StackRecursiveDoubleLinkedListImpl.java([IMPLEMENTAÇÃO PARCIAL DE UMA PILHA USANDO UMA LISTA DUPLA RECURSIVA](#))
2. Envie esse arquivo com sua solução para o sistema de submissão e verifique que o contador de submissões será alterado.

### Observações finais:

- **A interpretação do exercício faz parte da atividade.**
- **A atividade é individual. A conversa entre alunos é proibida.**
- **É proibido coletar códigos prontos e adaptar. Implemente as questões. Isso é para seu aprendizado.**
- **Caso você observe qualquer problema no sistema de submissão, contate o professor imediatamente.**
- **Se você não compactar o arquivo seguindo a estrutura de diretórios a compilação não terá sucesso e o sistema mostrará isso. Erro de compactação será de responsabilidade do aluno. O professor não ajudará o aluno nesse item. É só seguir as instruções deste arquivo.**