Instituto Federal de Educação, Ciência e Tecnologia de Sergipe – IFS		**Campus**: Itabaiana
Curso: Ciência da Computação	**Disciplina:** Estruturas de Dados I	
Professor: Levi Mota	**Avaliação da unidade 1:** Lista linear	**Data:** __/___/____
Aluno:		**Nota**:

Trabalho Avaliativo

Orientações

Tema: Implementação de uma Lista de Presença de alunos com base em uma Estrutura de Dados tipo Lista, com operações avançadas.

Duração estimada: 4 a 6 horas de trabalho.
Peso: substitui a primeira avaliação bimestral.
Entrega: individual ou em dupla.
Linguagem: Java
Observações: seguir os padrões da disciplina (uso de arrays, não usar bibliotecas prontas de lista como ArrayList).

Descrição do projeto:

Desenvolver um sistema simples de **controle de presença** para uma turma de alunos, utilizando **uma lista sequencial implementada com arrays**.

O sistema deve permitir:

1. Cadastrar um aluno com nome e matrícula.
2. Marcar presença de alunos.
3. Remover um aluno da lista.
4. Consultar um aluno por matrícula.
5. Listar todos os alunos presentes.
6. Listar todos os alunos da turma em ordem de cadastro.
7. Limpar a turma.
8. Transferir alunos de uma turma para outra.

Requisitos Técnicos

1) Criar uma classe Aluno com atributos nome, matricula (String) e presente (boolean).
2) Criar uma classe Turma com os seguintes métodos:
   a) void adicionarAluno(Aluno aluno)
   b) boolean removerAluno(String matricula)
   c) Aluno buscarAluno(String matricula)
   d) boolean marcarPresenca(String matricula)
   e) Lista listarPresentes()
   f) limparTurma()
   g) transferirTurma(Turma outra)
   h) String toString() → lista formatada de alunos
3) Criar uma classe de estrutura de dados de Lista para armazenar os alunos dentro de Turma. A classe deve considerar a implementação prévia da Lista implementada nas aulas e exercícios anteriores, com as características:
   a) A classe deve se chamar ListaArranjo;
   b) Opcionalmente, pode implementar uma interface chamada Lista, com os seguintes métodos definidos:

```
i)   public void adiciona(Object element);
ii)  public void adiciona(int pos, Object element);
iii) public Object pega(int posicao);
iv)  public void remove(int posicao);
v)   public int tamanho();
vi)  public int busca(Object element);
```

   c) Deve armazenar as informações em arranjo inicialmente criado com 20 posições.
   d) Deve permitir alocação dinâmica de espaço, criando um novo arranjo caso seu tamanho seja extrapolado.
   e) Deve ser generalista (aceitar Object como elemento) mas não precisa ser parametrizada (usar generics), apesar de recomendável que seja.
   f) Deve trabalhar com atributo que guarda o seu tamanho para reduzir a complexidade de algumas operações.
   g) Deve acrescentar os métodos para as operações adicionais a seguir, que serão úteis para a classe Turma ou para a aplicação:
      i) Limpar a lista, ou seja, remover todos os elementos: limpa().
      ii) Inverte a ordem dos elementos da lista: inverte().
      iii) Copia o conteúdo de uma outra Lista passada como parâmetro para esta Lista, incluindo todos os elementos do outra nesta: adicionaTodos(Lista outra).
4) Criar uma classe Programa que exibe um menu, espera a opção do usuário e executa as operações disponíveis para a Turma.
   a) Use a classe Scanner para obter as entradas do usuário, conforme material disponibilizado "02.2 - Java - Saída e entrada de dados.pdf".

Critérios de Avaliação

Item	Pontos
Implementação correta das operações	8,0
Organização do código, uso de POO, nomes significativos	1,0
Criatividade ou funcionalidades extras (ordenar, persistir, etc.)	1,0
Total	**10,0**