

**Atividades iguais serão zeradas.**

**Exercícios que envolvam codificações iguais serão zerados.**

1. Defina cada um dos itens abaixo e apresente exemplos de aplicação:

- Recursividade direta
- Recursividade indireta
- Divisão e conquista
- Busca binária e busca sequencial

2. Implemente uma Lista de temperaturas em alocação dinâmica com todas as operações indicadas a seguir **e que utilize pelo menos um método recursivo.**

- verificar se a lista está vazia, retornando true se estiver vazia e false se não estiver;
- adicionar uma temperatura no início da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
- adicionar uma temperatura no final da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
- adicionar uma temperatura em determinada posição da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
- remover a temperatura do início da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- remover a temperatura do final da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- remover a temperatura de determinada posição da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- percorrer e apresentar cada um dos elementos da lista.
- **Explique como a recursividade foi aplicada em cada um dos métodos, detalhando se foi aplicada recursividade direta ou indireta.**

3. A classe Aluno possui atributos que são criados na classe concreta e acessados por meio dos métodos públicos da classe (getters e setters). Dado o cenário implemente uma Lista Encadeada **que utilize pelo menos um método recursivo.**

- Sejam criados em uma classe os atributos RA de tipo de dado inteiro, nome de tipo de dado String, turma do tipo de dado String e semestre do tipo de dado String. Estes atributos devem ser acessados por meio de seus métodos públicos (getters e setters).
- Implemente um método que adicione um aluno no final da lista
- Implemente um método que adicione um aluno no início da lista
- Implemente um método que adicione um aluno no meio da lista

- Implemente um método que remova um aluno no final da lista
- Implemente um método que remova um aluno no início da lista
- Implemente um método que remova um aluno no meio da lista
- Implemente um método que apresente cada um dos elementos da lista
- **Explique como a recursividade foi aplicada em cada um dos métodos, detalhando se foi aplicada recursividade direta ou indireta.**