

10,0

DESCRIÇÃO DO PROBLEMA

Você foi encarregado de construir um sistema de gerenciamento de alunos baseado na média final. Implemente as funções para as funcionalidades requeridas a seguir.

- obs1: Nenhuma das funções deve permitir que os dados de entrada sejam alterados.
- obs2: A nomenclatura das funções deve respeitar a utilizada na bateria de testes.
- obs3: A bateria de testes traz exemplo de entrada e saída para cada questão.
- obs4: Cada funcionalidade corretamente implementada e que respeite o paradigma funcional valerá 2,0 pts.

1. **Transformar a lista de strings em uma lista de registros.** Os dados dos alunos serão fornecidos em uma lista de strings, onde cada string contém uma frase informando o primeiro nome e a média do aluno. O padrão do texto pode variar bastante, mas o nome do aluno é o único dado que sempre começa com letra maiúscula e tem pelo menos duas letras e a média é o único dado numérico presente.
[A-Z2-] 2,0
2. **Adicionar um novo aluno.** Crie uma função que receba a lista de registros dos alunos, o nome de um novo aluno e a média desse novo aluno e o adicione à lista. 2,0
3. **Filtrar alunos aprovados.** Crie uma função que receba a lista de registros de alunos e um valor numérico representando a média mínima de aprovação da instituição. A função deve retornar uma nova lista contendo apenas os alunos que foram aprovados, ou seja, aqueles cuja média é igual ou superior à média de aprovação. O resultado deve ser uma lista de registros com o nome do aluno e sua média. 2,0
4. **Classificar os alunos em grupos de desempenho.** Crie uma função que classifique os alunos em diferentes grupos de desempenho com base nos valores das médias: "RUIM" (abaixo de 5.0), "REGULAR" (entre 5.0 e menor que 7.0), "BOM" (a partir de 7.0). O resultado deve ser uma lista de registros de alunos contendo um campo a mais chamado desempenho que guardará essas nomenclaturas citadas. 2,0
5. **Calcular a média geral das notas.** Crie uma função que receba a lista de registros de alunos e calcule a média geral das médias. O resultado deve ter apenas uma casa decimal. 2,0

BATERIA DE TESTES e SAÍDAS ESPERADAS

```
// Funcionalidade 1
console.log('\nFuncionalidade 1');
const inputStr = [
  "Jo tem média 7.8.",
  "6.2 é a média de Maria.",
  "A média de Pedro é 4.5.",
  "Sofia: média = 2.9."
];
const alunosProcessados = processarStringAlunos(inputStr);
console.log(alunosProcessados);
```

Nome do Aluno
Letra Maiúscula
Nota = X.X

```
// Saída esperada:
// [
//   { nome: 'Jo', media: 7.8 },
//   { nome: 'Maria', media: 6.2 },
//   { nome: 'Pedro', media: 4.5 },
//   { nome: 'Sofia', media: 2.9 }
// ]
```

```
// Funcionalidade 2
console.log('\nFuncionalidade 2');
const listaDeAlunos = [
  { nome: 'Jo', media: 7.8 },
  { nome: 'Maria', media: 6.2 },
  { nome: 'Pedro', media: 4.5 },
  { nome: 'Sofia', media: 2.9 }
];
const novoAluno = "Gabriel";
const novaMedia = 9.9;
console.log(adicionarAluno(listaDeAlunos, novoAluno, novaMedia));
```

2
7,8
()
4,5
2,9
21,4

21,4 | 4
15 5,35
20

```
// Saída esperada:
// [
//   { nome: 'Jo', media: 7.8 },
//   { nome: 'Maria', media: 6.2 },
//   { nome: 'Pedro', media: 4.5 },
//   { nome: 'Sofia', media: 2.9 },
//   { nome: 'Gabriel', media: 9.9 }
// ]
```

```
// Funcionalidade 3
console.log('\nFuncionalidade 3');
const instMedia = 5.0;
console.log(filtrarAprovados(listaDeAlunos, instMedia));
```

```
// Saída esperada:
```

```
// [
//     { nome: 'Jo', media: 7.8 },
//     { nome: 'Maria', media: 6.2 }
// ]

// Funcionalidade 4
console.log('\nFuncionalidade 4');
console.log(classificarAlunos(listaDeAlunos));

// Saída esperada:
// [
//     { nome: 'Jo', media: 7.8, desempenho: 'BOM' },
//     { nome: 'Maria', media: 6.2, desempenho: 'REGULAR' },
//     { nome: 'Pedro', media: 4.5, desempenho: 'RUIM' },
//     { nome: 'Sofia', media: 2.9, desempenho: 'RUIM' }
// ]

// Funcionalidade 5
console.log('\nFuncionalidade 5');
console.log(calcularMediaGeral(listaDeAlunos));

// Saída esperada:
// 5.3

// Prova de que a lista original não foi alterada
console.log('\nLista original');
console.log(listaDeAlunos);

// Saída esperada:
// [
//     { nome: 'Jo', media: 7.8 },
//     { nome: 'Maria', media: 6.2 },
//     { nome: 'Pedro', media: 4.5 },
//     { nome: 'Sofia', media: 2.9 }
// ]
```

//1.

```
const processarAlunos = (entrada) => entrada.map((aluno) => {  
  const nome = aluno.match(/[A-Z]\w+/) aluno ✓  
  const media = aluno.match(/(\d+(\.\d+)?) /) ✓  
  return {nome: nome[0], media: parseFloat(media[0])} ✓  
})
```

//2.

```
const adicionarAluno = (lista, alunoNovo, mediaNova) => {  
  return [...lista, {nome: alunoNovo, media: mediaNova}] ✓  
}
```

//3.

```
const filtrarAprovados = (lista, corte) => lista.filter((aluno) => aluno.media >= corte) ✓
```

//4.

```
const classificarAlunos = (lista) => lista.map((aluno) => {  
  return {...aluno, desempenho: {  
    if (aluno.media < 5.0) return "RUIM"  
    if (aluno.media >= 5.0 && aluno.media < 7.0) return "REGULAR"  
    else return "BOM"  
  }}  
})
```

//5.

```
const calcularMedia Geral = (lista) => (lista.reduce((acc, x) => acc + x.media, 0) / lista.length).toFixed(1) ✓
```