

CURSO TECNOLÓGICO SUPERIOR EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS

ESTRUTURA DE DADOS



Centro Paula Souza



GOVERNO DO ESTADO
DE SÃO PAULO

TATIANE SILVA CACIQUE

SISTEMA DE CADASTRO DE ALUNOS

FRANCA/SP

ABRIL/2024

Estrutura do Código:

1. Variáveis e Elementos HTML:

res: variável que armazena o elemento HTML com o ID "res".

form: variável que armazena o elemento HTML com o ID "form".

encerrado: variável que armazena o elemento HTML com o ID "encerrado".

nome: variável que armazena o elemento HTML com o ID "nome".

ra: variável que armazena o elemento HTML com o ID "RA".

idade: variável que armazena o elemento HTML com o ID "idade".

sexo: variável que armazena o elemento HTML com o ID "sexo".

media: variável que armazena o elemento HTML com o ID "media".

botao: variável que armazena o elemento HTML com o ID "btn-cadastrar".

opcao: variável que armazena o elemento HTML com o ID "opcao".

confirmar: variável que armazena o elemento HTML com o ID "confirmar".

users: array que armazena os dados dos alunos cadastrados.

```
1 let res = document.getElementById("res");
2 let form = document.getElementById("form");
3 let encerrado = document.getElementById("encerrado");
4 let nome = document.getElementById("nome");
5 let ra = document.getElementById("RA");
6 let idade = document.getElementById("idade");
7 let sexo = document.getElementById("sexo");
8 let media = document.getElementById("media");
9 let botao = document.getElementById("btn-cadastrar");
0 let opcao = document.getElementById("opcao");
1 let confirmar = document.getElementById("confirmar");
```

2. Funções:

- Opções ():

Função que direciona o usuário para diferentes páginas HTML de acordo com a opção selecionada no menu suspenso.

As opções disponíveis são:

Cadastrar novo aluno ("/cadastro.html").

Relatório de alunos por ordem crescente de nome ("/relatorioNomeCrescente.html").

Relatório de alunos por ordem decrescente de RA ("/relatorioDecrescente.html").

Relatório de alunos aprovados por ordem crescente de nome ("/relatórioCrescenteAprovados.html").

Encerrar o programa e limpar os dados armazenados ("/index.html").

```
13 function Opcoes() {
14     switch (opcao.value) {
15         case "1":
16             location.href = "/cadastro.html";
17             break;
18         case "2":
19             location.href = "/relatorioNomeCrescente.html";
20             break;
21         case "3":
22             location.href = "/relatorioDecrescente.html";
23             break;
24         case "4":
25             location.href = "/relatórioCrescenteAprovados.html";
26             break;
27         case "5":
28             form.innerHTML = "";
29             encerrado.innerHTML = "Programa Encerrado";
30             localStorage.clear();
31             break;
32     }
33 }
```

- Retornar ():

Função que redireciona o usuário para a página inicial ("/index.html").

```
35 let users = [];
36 function Retornar() {
37     location.href = "/index.html"
38 }
```

- Cadastrar ():

Função que cadastra um novo aluno no sistema.

Valida se o RA do aluno já existe no sistema.

Calcula a situação do aluno (aprovado ou reprovado) com base na média.

Armazena os dados do aluno no array users.

Exibe uma mensagem de sucesso ou erro no elemento res.

Limpa os campos do formulário após o cadastro.

```
43 function Cadastrar() {
44     let userPersonal = {
45         nome: nome.value,
46         ra: ra.value,
47         idade: idade.value,
48         sexo: sexo.value,
49         media: media.value,
50         resultado: "",
51     };
52
53     if (media.value >= 6) {
54         situacao = "APROVADO";
55         userPersonal.resultado = situacao;
56     } else {
57         situacao = "REPROVADO";
58         userPersonal.resultado = situacao;
59     }
60
61     if (usersCadastrados.some((user) => user.ra === userPersonal.ra)) {
62         res.innerHTML = "ERRO, já possui um usuário com esse RA";
63     } else {
64         usersCadastrados.push(userPersonal);
65
66         localStorage.setItem("users", JSON.stringify(usersCadastrados));
67
68         res.innerHTML = "";
69         usersCadastrados.forEach((user) => {
70             res.innerHTML = "Aluno cadastrado com sucesso!"
71         });
72     }
73
74     nome.value = '';
75     ra.value = '';
76     idade.value = '';
77     sexo.value = '';
78     media.value = '';
79 }
```

- bubbleSort():

Função que implementa o algoritmo de ordenação Bubble Sort.

Recebe um vetor e uma função de comparação como parâmetros.

Ordena o vetor de acordo com a função de comparação.

```

81  function bubbleSort(vetor, fnComp) {
82      (pass = 0), (comps = 0), (trocas = 0);
83      let trocou;
84
85      do {
86          pass++;
87          trocou = false;
88          for (let i = 0; i < vetor.length - 1; i++) {
89              comps++;
90              if (fnComp(vetor[i], vetor[i + 1])) {
91                  [vetor[i], vetor[i + 1]] = [vetor[i + 1], vetor[i]];
92                  trocou = true;
93                  trocas++;
94              }
95          }
96      } while (trocou);
97  }

```

- NomeCrescente():

Função de comparação que ordena os alunos por ordem crescente de nome.

```

99  function NomeCrescente(a, b) {
100      return a.nome > b.nome;
101  }

```

- RADecrescente():

Função de comparação que ordena os alunos por ordem decrescente de RA.

```

103  function RADecrescente(a, b) {
104      return a.ra < b.ra;
105  }

```

- RelatorioNomes():

Função que gera um relatório dos alunos por ordem crescente de nome.

```

109 function RelatorioNomes() {
110     bubbleSort(usersCadastrados, NomeCrescente);
111     let alunos = "";
112     usersCadastrados.forEach((user) => {
113         alunos += `Nome: ${user.nome}, RA: ${user.ra}, <br> Idade: ${user.idade}, Sexo: ${user.sexo}, Média: ${user.media}, Resultado: ${user.resul
114     });
115     nomes.innerHTML = "";
116     if (alunos == '') {
117         nomes.innerHTML = "Não há alunos cadastrados";
118     } else {
119         nomes.innerHTML = alunos;
120     }
121 }

```

- RelatorioRA():

Função que gera um relatório dos alunos por ordem decrescente de RA.

```

125 function RelatorioRA() {
126     bubbleSort(usersCadastrados, RADecrescente);
127     let ListaAlunos = "";
128     usersCadastrados.forEach((user) => {
129         ListaAlunos += `RA: ${user.ra}, Nome: ${user.nome},<br> Idade: ${user.idade}, Sexo: ${user.sexo}, Média: ${user.media}, Resultado: ${user.r
130     });
131     alunos.innerHTML = "";
132     if (ListaAlunos == '') {
133         alunos.innerHTML = "Não há alunos cadastrados";
134     } else {
135         alunos.innerHTML = ListaAlunos;
136     }
137 }

```

- RelatorioAprovados():

Função que gera um relatório dos alunos aprovados por ordem crescente de nome.

```

141 function RelatorioAprovados() {
142     bubbleSort(usersCadastrados, NomeCrescente);
143     let alunos = "";
144     usersCadastrados.forEach((user) => {
145         if (user.resultado == "APROVADO") {
146             alunos += `Nome: ${user.nome}, RA: ${user.ra}, <br> Idade: ${user.idade}, Sexo: ${user.sexo}, Média: ${user.media}, Resultado: ${user.res
147         }
148     });
149     nomesAprovados.innerHTML = "";
150     if (alunos == '') {
151         nomesAprovados.innerHTML = "Não há alunos cadastrados";
152     } else {
153         nomesAprovados.innerHTML = alunos;
154     }
155 }

```