



# Proposta:

Baseado nos conhecimentos fundamentais em HTML, CSS e JavaScript.

Houve uma proposta para desenvolvemos uma solução de um site que sirva como um repositório de atividades propostas e desenvolvidas em HTML, CSS e JavaScript servindo de base para futuras consultas quando houver necessidade ou também para ajudar aqueles que estejam em processo de aprendizagem.

```
<div>
  <div>
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
                                                                                        <div>
                                                                                          <div>
                                                                                            <div>
                                                                                              <div>
                                                                                                <div>
                                                                                                  <div>
                                                                                                    <div>
                                                                                                      <div>
                                                                                                        <div>
                                                                                                          <div>
                                                                                                            <div>
                                                                                                              <div>
                                                                                                                <div>
                                                                                                                  <div>
                                                                                                                    <div>
                                                                                                                      <div>
                                                                                                                        <div>
                                                                                                                          <div>
                                                                                                                            <div>
                                                                                                                              <div>
                                                                                                                                <div>
                                                                                                                                  <div>
                                                                                                                                    <div>
                                                                                                                                      <div>
                                                                                                                                        <div>
                                                                                                                                          <div>
                                                                                                                                            <div>
                                                                                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```



# Estruturas condicionais:

SE

CASO

```
16
17     string sInput;
18     int iLength, iN;
19     double dblTemp;
20     bool again = true;
21
22     while (again) {
23         iN = -1;
24         again = false;
25         getline(cin, sInput);
26         system("cls");
27         stringstream(sInput) >> dblTemp;
28         iLength = sInput.length();
29         if (iLength < 4) {
30             again = true;
31             continue;
32         } else if (sInput[iLength - 3] != '.') {
33             again = true;
34             continue;
35         }
36     }
37 }
```



# Estruturas condicionais

## “SE”:

ATIVIDADE 1;

ATIVIDADE 2;

```
16
17     string sInput;
18     int iLength, iN;
19     double dblTemp;
20     bool again = true;
21
22     while (again) {
23         iN = -1;
24         again = false;
25         getline(cin, sInput);
26         system("cls");
27         stringstream(sInput) >> dblTemp;
28         iLength = sInput.length();
29         if (iLength < 4) {
30             again = true;
31             continue;
32         } else if (sInput[iLength - 3] != '.') {
33             again = true;
34             continue;
35         }
36     }
37 }
```



# Estruturas condicionais “CASO”:

ATIVIDADE 1;

ATIVIDADE 2;



```
17     string sInput;
18     int iLength, iN;
19     double dblTemp;
20     bool again = true;
21
22     while (again) {
23         iN = -1;
24         again = false;
25         getline(cin, sInput);
26         system("cls");
27         stringstream(sInput) >> dblTemp;
28         iLength = sInput.length();
29         if (iLength < 4) {
30             again = true;
31             continue;
32         } else if (sInput[iLength - 3] != '.') {
33             again = true;
34             continue;
35         }
36     }
37 }
```



# Estruturas de Repetição:

PARA  
FAÇA ENQUANTO  
ENQUANTO

```
16  
17     string sInput;  
18     int iLength, iN;  
19     double dblTemp;  
20     bool again = true;  
21  
22     while (again) {  
23         iN = -1;  
24         again = false;  
25         getline(cin, sInput);  
26         system("cls");  
27         stringstream(sInput) >> dblTemp;  
28         iLength = sInput.length();  
29         if (iLength < 4) {  
30             again = true;  
31             continue;  
32         } else if (sInput[iLength - 3] != '.') {  
33             again = true;  
34             continue;  
35         } else {  
36             again = false;  
37         }  
38     }
```



# Estruturas de Repetição “PARA”:

EXEMPLO;

ATIVIDADE 1;



```
17     string sInput;
18     int iLength, iN;
19     double dblTemp;
20     bool again = true;
21
22     while (again) {
23         iN = -1;
24         again = false;
25         getline(cin, sInput);
26         system("cls");
27         stringstream(sInput) >> dblTemp;
28         iLength = sInput.length();
29         if (iLength < 4) {
30             again = true;
31             continue;
32         } else if (sInput[iLength - 3] != '.') {
33             again = true;
34             continue;
35         }
36     }
37 }
```



# Estruturas de Repetição

## “Faça Enquanto”:

EXEMPLO;

ATIVIDADE 1;

```
17 string sInput;
18 int iLength, iN;
19 double dblTemp;
20 bool again = true;
21
22 while (again) {
23     iN = -1;
24     again = false;
25     getline(cin, sInput);
26     system("cls");
27     stringstream(sInput) >> dblTemp;
28     iLength = sInput.length();
29     if (iLength < 4) {
30         again = true;
31         continue;
32     } else if (sInput[iLength - 3] != '.') {
33         again = true;
34         continue;
35     }
36 }
```



# Estruturas de Repetição “Enquanto”:

EXEMPLO;

ATIVIDADE 1;

```
17     string sInput;
18     int iLength, iN;
19     double dblTemp;
20     bool again = true;
21
22     while (again) {
23         iN = -1;
24         again = false;
25         getline(cin, sInput);
26         system("cls");
27         stringstream(sInput) >> dblTemp;
28         iLength = sInput.length();
29         if (iLength < 4) {
30             again = true;
31             continue;
32         } else if (sInput[iLength - 3] != '.') {
33             again = true;
34             continue;
35         }
36     }
37 }
```



# VETORES:

UNIDIMENSIONAL  
BIDIMENSIONAL

```
17 string sInput;
18 int iLength, iN;
19 double dblTemp;
20 bool again = true;
21
22 while (again) {
23     iN = -1;
24     again = false;
25     getline(cin, sInput);
26     system("cls");
27     stringstream(sInput) >> dblTemp;
28     iLength = sInput.length();
29     if (iLength < 4) {
30         again = true;
31         continue;
32     } else if (sInput[iLength - 3] != '.') {
33         again = true;
34         continue;
35     }
36 }
```



# VETOR “UNIDIMENSIO NAL”:

EXEMPLO;

ATIVIDADE 1;

```
17 string sInput;
18 int iLength, iN;
19 double dblTemp;
20 bool again = true;
21
22 while (again) {
23     iN = -1;
24     again = false;
25     getline(cin, sInput);
26     system("cls");
27     stringstream(sInput) >> dblTemp;
28     iLength = sInput.length();
29     if (iLength < 4) {
30         again = true;
31         continue;
32     } else if (sInput[iLength - 3] != '.') {
33         again = true;
34         continue;
35     }
36 }
```



# VETOR “BIDIMENSIONAL”:

EXEMPLO;

ATIVIDADE 1;

```
17 string sInput;
18 int iLength, iN;
19 double dblTemp;
20 bool again = true;
21
22 while (again) {
23     iN = -1;
24     again = false;
25     getline(cin, sInput);
26     system("cls");
27     stringstream(sInput) >> dblTemp;
28     iLength = sInput.length();
29     if (iLength < 4) {
30         again = true;
31         continue;
32     } else if (sInput[iLength - 3] != '.') {
33         again = true;
34         continue;
35     }
36 }
```