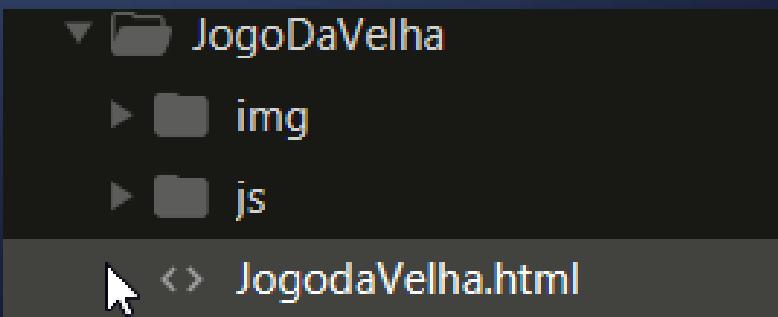




Projeto Jogo da Velha...

• Estrutura do projeto...

- Criamos uma nova pasta no Sublime Text com nome de “JogoDaVelha”
- Dentro dela vamos ter uma pasta com nome de “img” e outra pasta com nome de “js”
- Na pasta “img” Vamos colocar as imagens necessário para nosso projetos que vão estar disponíveis para download no link → [IMAGENS](#).
- Na pasta “js” vai estar o arquivo do jQuery disponível no link → [JQUERY](#).
- Dentro da pasta “JogoDaVelha” Teremos nossa página html com nome de “JogoDaVelha.html”



• Estrutura padrão do arquivo JogoDaVelha.html

```
◀ ▶   JogodaVelha.html ×  
1  <!DOCTYPE html>  
2  <html lang="pt-br">  
3      <head>  
4          <meta charset="UTF-8">  
5  
6          <title>Jogo da Velha</title>  
7          <script src="js/jquery-3.3.1.js"></script>  
8      </head>  
9  
10     <body>  
11  
12     </body>  
13  
14  </html>
```

- Primeiro vamos criar a tela inicial do nosso jogo;
- Uma tabela para inserir o nome de cada jogador e ter o botão de play para iniciar o jogo;

```
<body>
    <div id="inicio" align="center">
        <table border="1" width="600px">
            <tbody>
                <tr>
                    <td colspan="3" align="center"><h1>Jogo da Velha</h1></td>
                </tr>
            </tbody>
        </table>
    </div>

```

Jogo da Velha

```
<tr>

    <td align="center">
         Imagen Jogador 1
        <br>
        <input type="text" maxlength="10" placeholder="Nome Jogador 1" style="text-align: center; border: 0px" />
    </td> Nome do Jogador com máx. 10 caracteres Formatações de estilo do campo
    <td align="center">
         Imagen Botão Play
    </td>
    <td align="center">
         Imagen Jogador 2
        <br>
        <input type="text" maxlength="10" placeholder="Nome Jogador 2" style="text-align: center; border: 0px" />
    </td> Texto que aparece dentro do campo do nome
</tr>
</tbody>
</table>
</div>
```

- Começamos agora a construção da tela do jogo;
 - Criamos mais 3 divs.
 - Jogador 1
 - Tabuleiro do Jogo
 - Jogador 2
 - Utilizamos o estilo "display: inline-block" para deixar as 3 divs uma no lado da outra
- ```
</tr>
</tbody>
</table>
</div>

<div id="jogo" style="margin-top: 50px; text-align:center">
 <div style="display: inline-block">

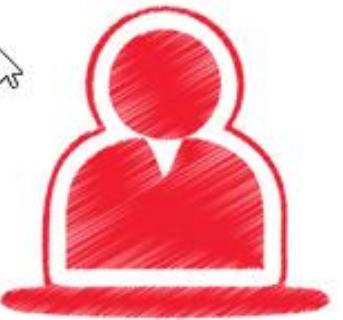
 </div>

 <div style="display: inline-block">

 </div>
</div>

</body>
</html>
```

# Jogo da Velha



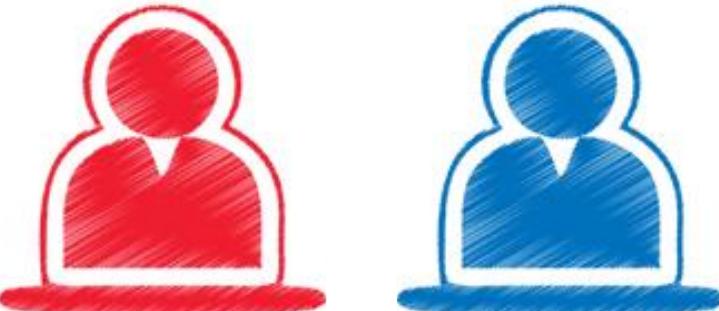
Nome Jogador 1



*Jogo da Velha*  
**Técnico em Informática**



Nome Jogador 2



Vamos criar agora a grid  
que ficará o jogo

```

</div>

<div style="display: inline-block">
 <table border="1">
 <tr>
 <td></td>
 <td></td>
 <td></td>
 </tr>
 <tr>
 <td></td>
 <td></td>
 <td></td>
 </tr>
 <tr>
 <td></td>
 <td></td>
 <td></td>
 </tr>
 <tr>
 <td></td>
 <td></td>
 <td></td>
 </tr>
 </table>
</div>
```

# Jogo da Velha



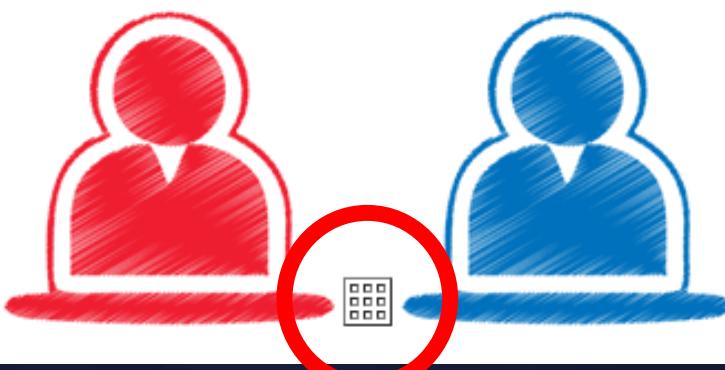
Nome Jogador 1



*Jogo da Velha*  
**Técnico em Informática**



Nome Jogador 2



- Dentro de cada <td> definimos uma altura e largura para organizar o tabuleiro do jogo.
- Criamos também uma borda personalizada igual utilizada no jogo da velha.

## LINHA 1

```
<div style="display: inline-block">
 <table border="0">
 <tr>
 <td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
 <td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
 <td style="border-bottom: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
 </tr>
```

# LINHA 2

```
<tr>
 <td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
 <td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
 <td style="border-bottom: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
</tr>
```

# LINHA 3

```
<tr>
 <td style="border-right: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
 <td style="border-right: solid 1px red;">
 <div style="width:100px; height:100px;"></div>
 </td>
 <td>
 <div style="width:100px; height:100px;"></div>
 </td>
</tr>
</table>
</div>
```

# Jogo da Velha



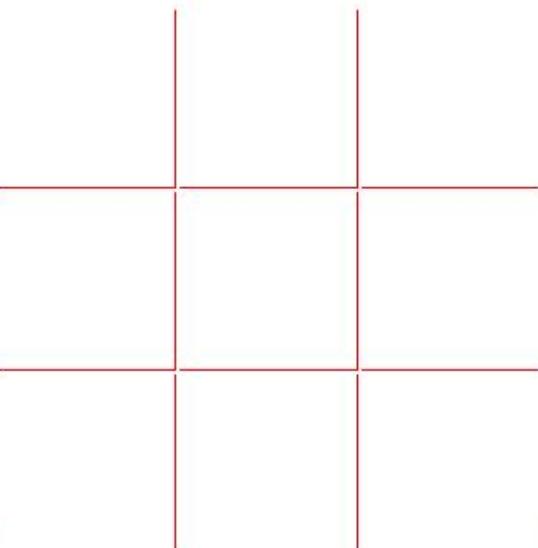
Nome Jogador 1



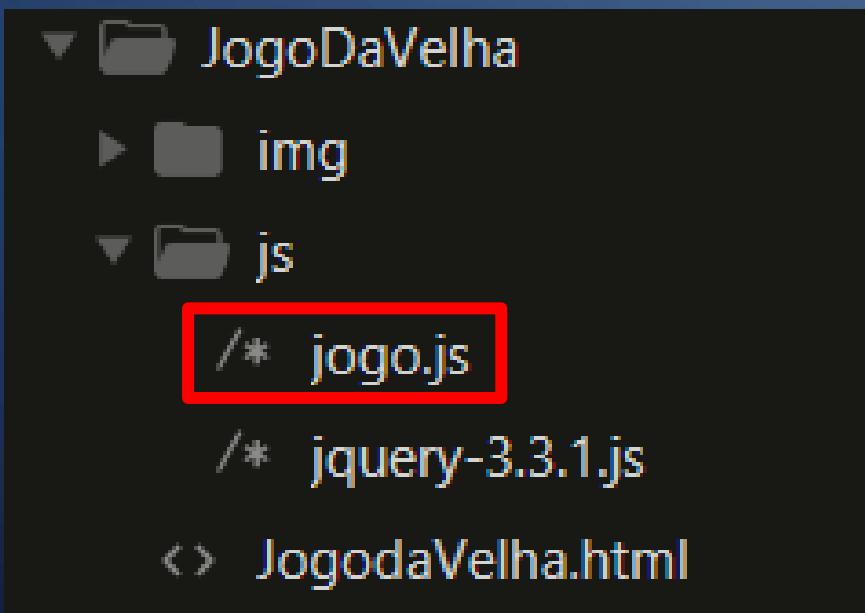
*Jogo da Velha*  
**Técnico em Informática**



Nome Jogador 2



- Igual ao projeto do jogo matar os patos, vamos criar um outro arquivo “jogo.js” para separar toda a programação que vamos realizar.



```
<title>Jogo da Velha</title>
<script src="js/jquery-3.3.1.js"></script>
<script src="js/jogo.js"></script>
</head>
```

- A programação do jogo inicia quando clicamos no botão PLAY, e para capturar este click vamos precisar definir um id para o botão.
- Realizamos uma programação de teste no arquivo “jogo.js” para verificar se está funcionando o botão de play.

```
</td>

<td align="center">

</td>
<td align="center">
```



The screenshot shows a browser window with two tabs: "JogodaVelha.html" and "jogo.js". The "jogo.js" tab contains the following JavaScript code:

```
$(document).ready(function(){
 $('#btn-play').click(function(){
 alert('Iniciar Jogo');
 });
});
```

An alert dialog box is displayed, containing the message "Iniciar Jogo" with an "OK" button. The "jogo.js" tab has a status bar at the bottom that says "Essa página diz Iniciar Jogo".

- Precisamos capturar os nomes que foram digitados para cada jogador na tela inicial do jogo e mostrar na parte do tabuleiro.
- Vamos definir um id para cada nome, assim vamos conseguir capturar as informações que foram digitadas.

```
<input id="nomejogador1" type="text" maxlength="10" placeholder="Nome Jogador 1" style="text-align:
center; border: 0px" />
```

```
<input id="nomejogador2" type="text" maxlength="10" placeholder="Nome Jogador 2" style="text-align:
center; border: 0px" />
```

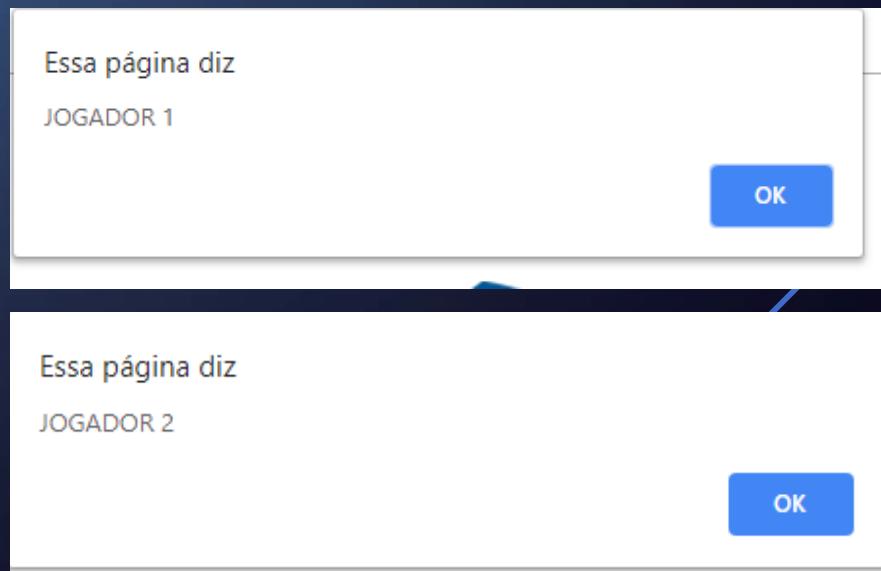
- Para testar se os nomes estão sendo capturados, vamos criar dois alerts para verificar.

```
$(document).ready(function(){

 $('#btn-play').click(function(){

 alert($('#nomejogador1').val());
 alert($('#nomejogador2').val());

 });
});
```



- Faltar criar uma verificação que obriga a digitação dos nomes para os jogadores.

```
$('#btn-play').click(function(){

 //Verifica se algum nome foi digitado
 if ($('#nomejogador1').val() == ''){
 alert('Digite um nome para o Jogador 1');
 return false;
 }

 if ($('#nomejogador2').val() == ''){
 alert('Digite um nome para o Jogador 2');
 return false;
 }

 alert($('#nomejogador1').val());
 alert($('#nomejogador2').val());
```

If – Faz a verificação se o campo está vazio. Se estiver mostra a mensagem que precisar preencher o nome do jogador.

- Depois da verificação se os nomes foram digitados, vamos definir que eles sejam mostrados em baixo de cada uma das imagens do jogador.

1. Em `JogodaVelha.html` vamos criar um `<span>` onde o nome será mostrado em baixo de cada uma das imagens;
2. Vamos definir um “`id`” para identificar cada `<span>`;
3. No `jogo.js` vamos codificar a função para chamar o nome dentro do `<span>`.

```
<div id="jogo" style="margin-top: 50px; text-align:center">
 <div style="display: inline-block">

 </div>
```

```
</div>

<div style="display: inline-block">

</div>
```

```
$(document).ready(function(){

 $('#btn-play').click(function(){

 //Verifica se algum nome foi digitado
 if ($('#nomejogador1').val() == ''){
 alert('Digite um nome para o Jogador 1');
 return false;
 }

 if ($('#nomejogador2').val() == ''){
 alert('Digite um nome para o Jogador 2');
 return false;
 }

 //Mostrar os nome na tela do jogo
 $('#nome-jogador-1').html($('#nomejogador1').val());
 $('#nome-jogador-2').html($('#nomejogador2').val());

 });

});
```

**Os dois alerts criados anteriormente podem ser apagados, pois já verificamos que está funcionando a captura dos nomes.**

- Nosso Grid deve começar desabilitado e só liberar para jogar depois que os jogadores inserirem seus nomes e clicarem no Play.
- Depois disso ele desabilita a opção de mudar o nome dos jogadores.

```
JogodaVelha.html x

<div id="jogo" style="display:none; margin-top: 50px; text-align:center">
 <div style="display: inline-block">

 </div>
```

```
//Mostrar os nome na tela do jogo
$('#nome-jogador-1').html($('#nomejogador1').val());
$('#nome-jogador-2').html($('#nomejogador2').val());
```

```
//Controla a visualização da tela do jogo
```

```
$('#inicio').hide();
```

```
$('#jogo').show();
```

```
});
```

```
});
```

- Todas as divs do nosso tabuleiro terão uma classe em comum, para podermos capturar o clique das jogadas.

```
<td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div class="jogada" style="width:100px; height:100px;"></div>
</td>

<td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div class="jogada" style="width:100px; height:100px;"></div>
</td>

<td style="border-bottom: solid 1px red;">
 <div class="jogada" style="width:100px; height:100px;"></div>
</td>
```

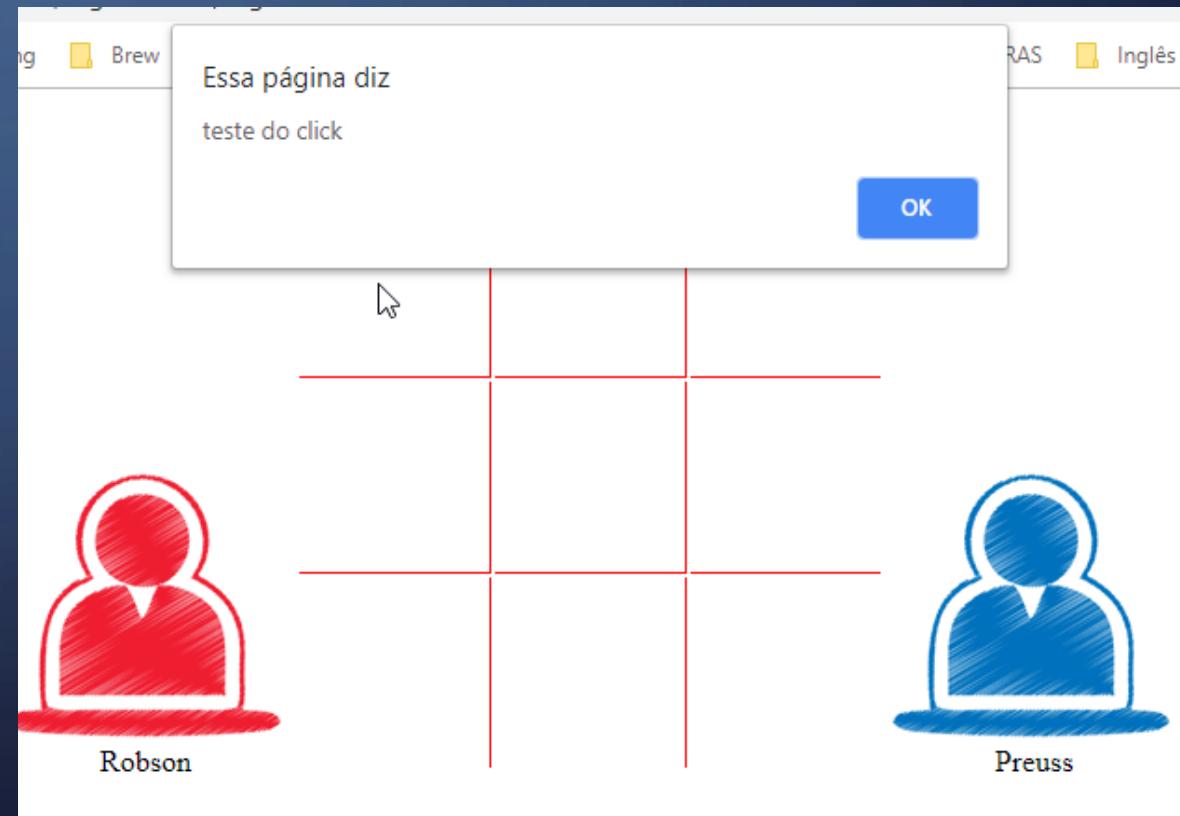
- APLICAR “class=“jogada” EM TODAS AS DIVS QUE FAZEM PARTE DO TABULEIRO DO JOGO.

- Dentro do `jogo.js` vamos criar a função que vai capturar o click dentro da nossa grid através da classe que foi definida.
- Criamos uma alert para testar se todos os campos da grid estão sendo capturado com o click.

```
//Controla a visualização da tela do jogo
$('#inicio').hide();
$('#jogo').show();

});

$('.jogada').click(function(){
 alert('teste do click');
});
});
```



- Pra identificar em qual dos campos o jogador clicou, precisamos definir um identificador único para cada campo (id).

A1	A2	A3
B1	B2	B3
C1	C2	C3

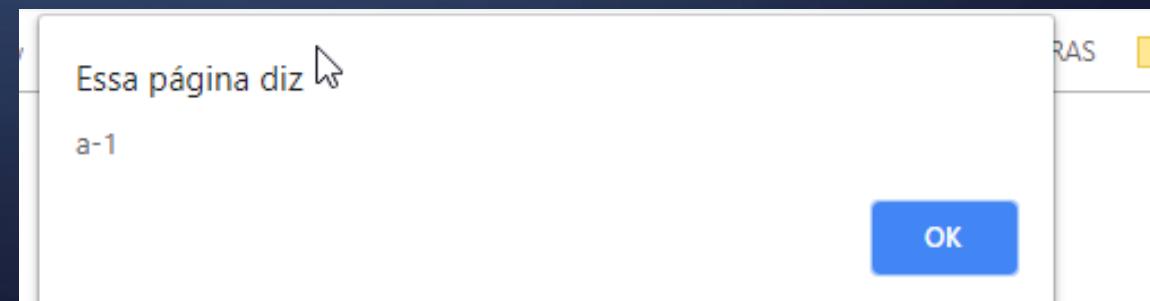
```
<td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div class="jogada" id="a-1" style="width:100px; height:100px;"></div>
</td>
<td style="border-bottom: solid 1px red; border-right: solid 1px red;">
 <div class="jogada" id="a-2" style="width:100px; height:100px;"></div>
</td>
<td style="border-bottom: solid 1px red;">
 <div class="jogada" id="a-3" style="width:100px; height:100px;"></div>
</td>
```

**CRIAR PARA TODAS  
AS LINHAS DA GRID**

- Agora vamos recuperar o valor individual de cada DIV, assim podemos controlar individualmente cada elemento.
- Criamos uma variável `id_campo` e atribuímos o id do elemento. O "this" faz referência ao elemento do contexto do click.

```
$('.jogada').click(function(){

 var id_campo = this.id;
 $('#'+id_campo).off(); //DESABILITA O CAMPO PARA NÃO PODER MUDAR DE OBJETO
 jogada(id_campo);
});
```



- No escopo mais alto do nosso código, criamos uma variável que vai a todo momento controlar o escopo do jogo e as jogadas. Teremos também um Array que vai controlar as posições 1, 2 e 3 para cada jogada



```
var rodada = 1;
var matriz_jogo = Array(3);

$(document).ready(function(){

 $('#btn-play').click(function(){

 //Verifica se algum nome foi digi
```

- Vamos precisar criar uma função que será capaz de entender onde foi clicado e atribuir uma imagem.

```
function jogada(id){
 var icone = '';
 var ponto = 0;

 if((rodada % 2) == 1){
 icone = 'url("img/m1.png")';
 ponto = -1;
 } else {
 icone = 'url("img/m2.png")';
 ponto = 1;
 }
 rodada++;

 $('#'+id).css('background-image', icone);
}
```

- Agora vamos organizar a matriz que controla todo o jogo. O jogo da velha pode ser dividido em linhas e colunas, com a combinação de uma linha e uma coluna nos temos uma coordenada.

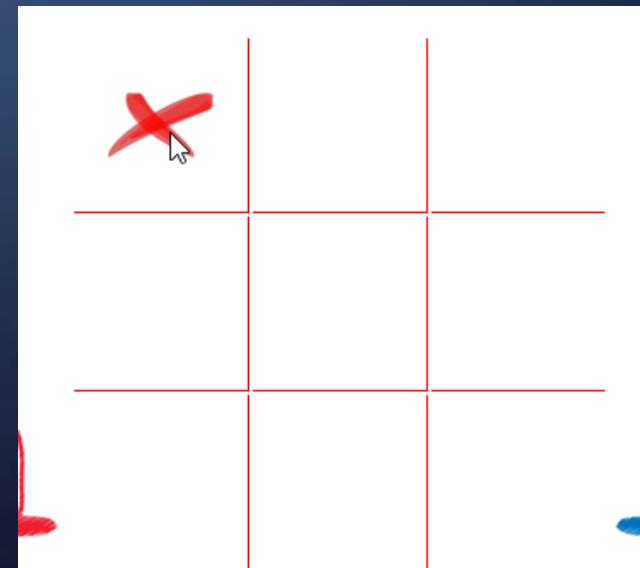
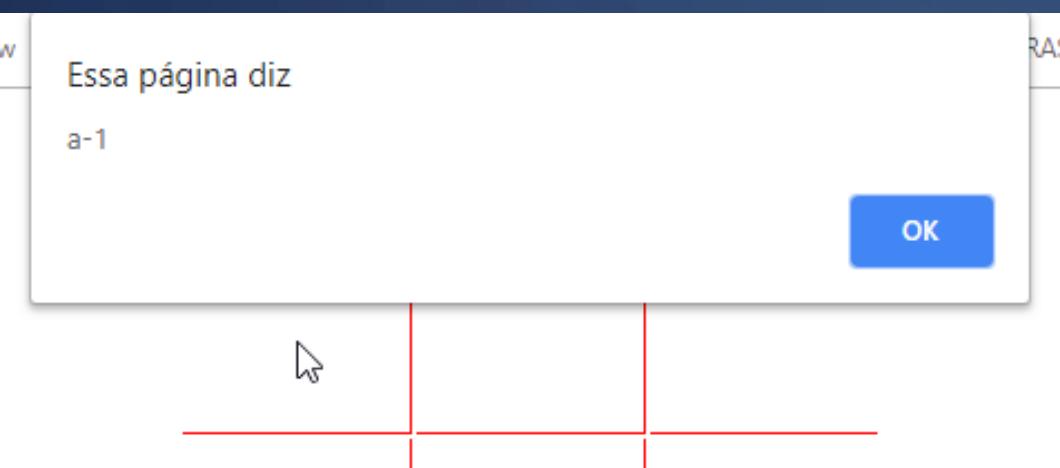
	1	2	3
A	a-1	a-2	a-3
B	b-1	b-2	b-3
C	c-1	c-2	c-3

- Temos posições que são “A, B, C” – “1,2,3” com subposições “a1, a2, a3...”

```
1 var rodada = 1;
2 var matriz_jogo = Array(3);
3
4 matriz_jogo['a'] = Array(3);
5 matriz_jogo['b'] = Array(3);
6 matriz_jogo['c'] = Array(3);
7
8 matriz_jogo['a'][1] = 0;
9 matriz_jogo['a'][2] = 0;
10 matriz_jogo['a'][3] = 0;
11
12 matriz_jogo['b'][1] = 0;
13 matriz_jogo['b'][2] = 0;
14 matriz_jogo['b'][3] = 0;
15
16 matriz_jogo['c'][1] = 0;
17 matriz_jogo['c'][2] = 0;
18 matriz_jogo['c'][3] = 0;
19
20 $(document).ready(function(){
21 $('#btn-play').click(function(){
```

- Para testar o funcionamento do código até o momento vamos criar um alert.

```
$('#'+id).css('background-image', icone);
 alert(id);|
});
```



- Até o momento a programação traz a coordenada certa de qual elemento estamos clicando. Precisamos realizar agora a quebra deste ID para poder inserir os pontos de cada jogador.
- Para realizar esta quebra, utilizando a função split que através de um carácter específico consegue dividir nossa informação e armazenar em um Array.

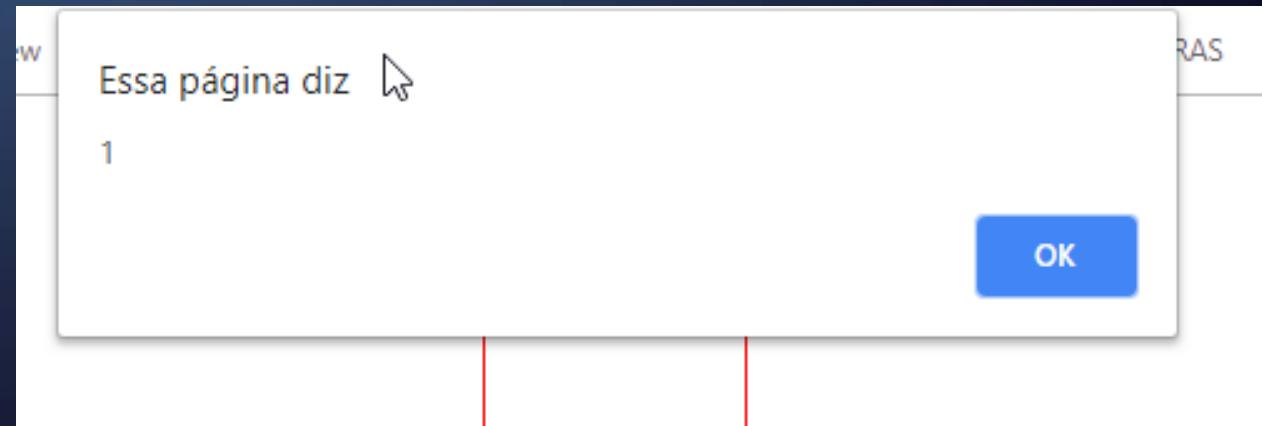
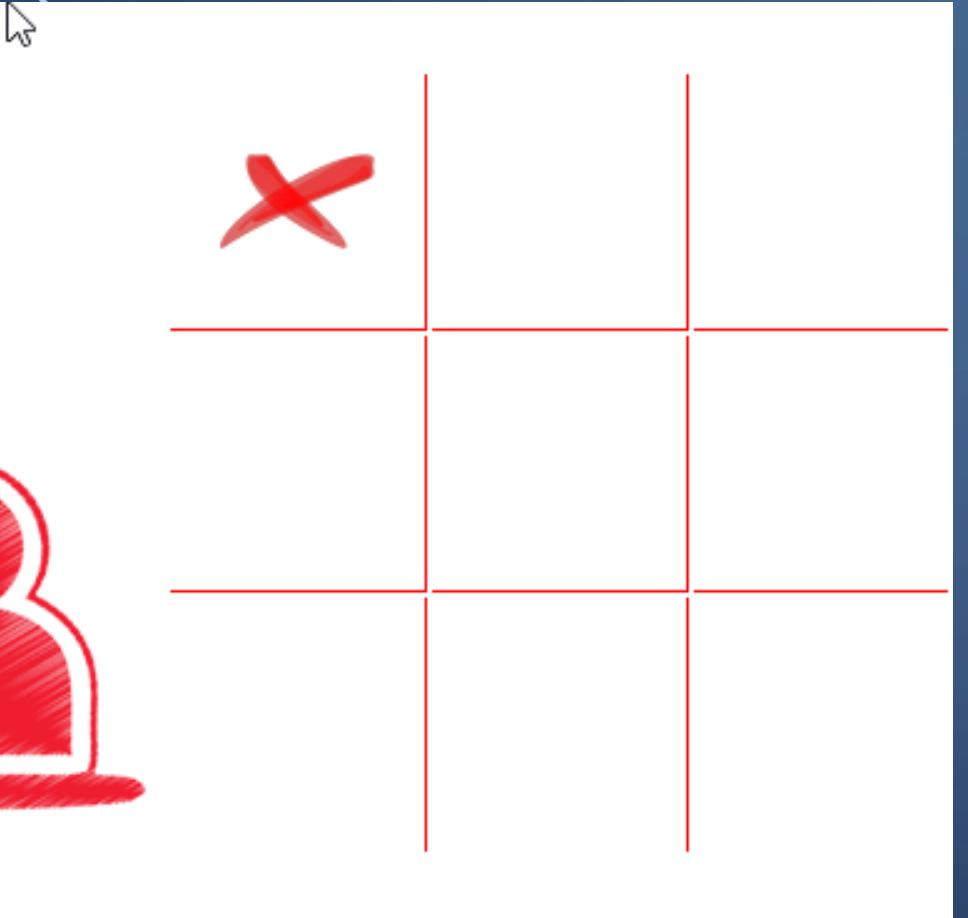
```
$('#'+id).css('background-image', icone);

var linha_coluna = id.split('-');

alert(linha_coluna[0]);
alert(linha_coluna[1]);

});
```

Como não precisamos forçar a chave do Array o próprio JavaScript insere a informação de 0 e nesta posição ele vai armazenar o valor da linha e na posição 1 ele armazena a posição da coluna.



- Já temos as informações separadas para inserção da pontuação dentro da matriz do jogo
- Primeira chave é linha e a segunda chave é coluna e para finalizar atribuímos o “ponto”.

```
$('#'+id).css('background-image', icone);

var linha_coluna = id.split('-');

matrix_jogo[linha_coluna[0]][linha_coluna[1]] = ponto;
```

• Para testar o funcionamento vamos inserir um “console.log” na matriz do jogo.

```
$('#'+id).css('background-image', icone);

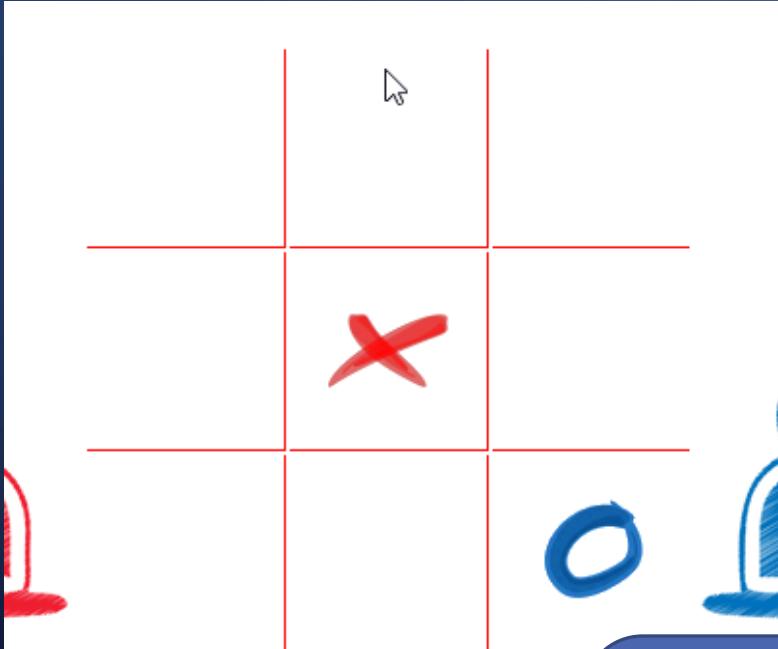
var linha_coluna = id.split('-');

matrix_jogo[linha_coluna[0]][linha_coluna[1]] = ponto;

console.log(matrix_jogo);

});
```

- Com o Google Chrome aberto digitamos “CTRL+SHIFT+I” e vamos na aba “Console” para analisar a matriz do jogo.
- Quando definimos um objeto na matriz o console mostra as informações.



```
▼ (3) [empty × 3, a: Array(4), b: Array(4), c: Array(4)] jogo.js:70
 ► a: (4) [empty, 0, 0, 0]
 ► b: (4) [empty, 0, -1, 0]
 ► c: (4) [empty, 0, 0, 0]
 length: 3
 ► __proto__: Array(0)

▼ (3) [empty × 3, a: Array(4), b: Array(4), c: Array(4)] jogo.js:70
 ► a: (4) [empty, 0, 0, 0]
 ► b: (4) [empty, 0, -1, 0]
 ► c: (4) [empty, 0, 0, 1]
 length: 3
 ► __proto__: Array(0)
```

Para diferenciar os dois jogadores.... Quando o JOGADOR 1 está jogando ele coloca “-1” e quando o JOGADOR 2 joga ele coloca “1”

- Agora vamos realizar a verificação que faz a checagem a cada rodada para ver se houve um ganhador ou não.

```
matriz_jogo[linha_coluna[0]][linha_coluna[1]] = ponto;
```

```
verifica_pontuacao();
```

```
}
```

## VERIFICAÇÃO NA HORIZONTAL

```
I
function verifica_pontuacao(){
 //Verifica na horizontal
 var pontos = 0;
 for(var i = 1; i <=3; i++){
 pontos = pontos + matriz_jogo['a'][i];
 }
 ganhador(pontos);

 pontos = 0;
 for(var i = 1; i <=3; i++){
 pontos = pontos + matriz_jogo['b'][i];
 }
 ganhador(pontos);

 pontos = 0;
 for(var i = 1; i <=3; i++){
 pontos = pontos + matriz_jogo['c'][i];
 }
 ganhador(pontos);

}
```

## VERIFICAÇÃO NA VERTICAL

```
pontos = 0;
for(var i = 1; i <=3; i++){
 pontos = pontos + matriz_jogo['c'][i];
}
ganhador(pontos);

//Verifica na Vertical
for(var l = 1; l <= 3; l++){
 pontos = 0;
 pontos += matriz_jogo['a'][l];
 pontos += matriz_jogo['b'][l];
 pontos += matriz_jogo['c'][l];

 ganhador(pontos);
}
```

## VERIFICAÇÃO NA DIAGONAL

```
//Verifica na Diagonal
pontos = 0;
pontos = matriz_jogo['a'][1] + matriz_jogo['b'][2] + matriz_jogo['c'][3];
ganhador(pontos);

pontos = 0;
pontos = matriz_jogo['a'][3] + matriz_jogo['b'][2] + matriz_jogo['c'][1];
ganhador(pontos);
}
```

- Criamos agora o alert do ganhador do Jogo.

```
function ganhador(pontos){
 if(pontos == -3){
 var jogador1 = $('#nomejogador1').val();
 alert(jogador1 + ' é o VENCEDOR!');
 //Desabilita jogo depois de um ganhador
 $('.jogada').off();

 } else if(pontos == 3){
 var jogador2 = $('#nomejogador2').val();
 alert(jogador2 + ' é o VENCEDOR!');
 //Desabilita jogo depois de um ganhador
 $('.jogada').off();
 }
}
});
```