

04

## Jogo - PARTE 2

### Transcrição

Nosso jogo já consegue chegar até a segunda etapa. Sabemos que é nesta etapa que o jogador informa o seu chute e que nosso jogo tem que ser capaz de preencher corretamente seu array `lacunas` com a letra informada, inclusive exibir o próximo sprite da forca caso ele erre. Nesse sentido, nossa função `criaJogo()` precisará receber como parâmetro um sprite. Vamos aproveitar e adicionar o esqueleto da função `processaChute`, aquela que lerá o chute do jogador e atualizará `lacunas`:

```
// recebe sprite como parâmetro
var criaJogo = function (sprite) {

    var etapa = 1;
    var lacunas = [];
    var palavraSecreta = '_';

    var criaLacunas = function () {

        for (let i = 0; i < palavraSecreta.length; i++) {
            lacunas.push('_');
        }
    };

    var proximaEtapa = function () {

        etapa = 2;
    };

    var setPalavraSecreta = function (palavra) {

        palavraSecreta = palavra;
        criaLacunas();
        proximaEtapa();
    };

    var getLacunas = function () {

        return lacunas;
    };

    var getEtapa = function () {

        return etapa;
    };

    // preencher lacuna ou exibe o próximo sprite. Retorna true ou false caso o jogador tenha acertado
    var processaChute = function() {
        console.log('falta implementar');

    };

    // adicionou na propriedade do objeto
}
```

```
        return {
            setPalavraSecreta: setPalavraSecreta,
            getLacunas: getLacunas,
            getEtapa: getEtapa,
            processaChute: processaChute
        }
    };

```

## Separando dependências

Talvez você esteja se perguntando porque eu não chamei `criaSprite('.sprite')` dentro de `criaJogo()` evitando assim passar um parâmetro para esta função. Essa abordagem permite a troca do sprite utilizado pela aplicação. Hoje, usamos o sprite de uma forca, talvez alguém queria usar outro sprite com uma conotação menos sombria. Sendo assim, quem criar um sprite com as mesmas funções que definimos ao criar nosso primeiro sprite poderá utilizá-lo em nosso jogo sem ter que mexer em sua lógica.

Mais uma vez, abrirei o projeto com esta parte pronta e os testes que devem ser realizados para que possamos ver qual será o resultado esperado:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/sprite.css">
    <link rel="stylesheet" href="css/lacuna.css">
    <title>Jogo da Forca</title>
</head>
<body>
    <div class="jumbotron">
        <h2 class="text-center">Jogo da forca</h2>
    </div>
    <div class="container">
        <div class="sprite"></div>
        <ul class="lacunas text-center"></ul>

        <div class="text-center">
            <input id="entrada"
                class="entrada"
                placeholder="Palavra secreta" autofocus>
        </div>
    </div>

    <script src="js/jquery.min.js"></script>
    <script src="js/sprite.js"></script>
    <script src="js/jogo.js"></script>
    <script>
        var jogo = criaJogo(createSprite('.sprite'));
        jogo.setPalavraSecreta('calopsita');
        console.log(jogo.getLacunas()); // ['', '', '', '', '', '', '', '']
        jogo.processaChute('a');
        console.log(jogo.getLacunas()); // ['', 'a', '', '', '', '', 'a']
        jogo.processaChute('c');
    </script>

```

```
console.log(jogo.getLacunas()); // ['c', 'a', '', '', '', '', '', 'a']
jogo.processaChute('z'); // próximo sprite deve ser exibido
</script>
</body>
</html>
```