

# Lógica de Programação: comece em lógica com o jogo Pong e JavaScript

## ◆ Questão 1 (Scratch)

No processo de criação dos atores, é necessário pintar o ator. Para a bolinha, precisamos fazer um círculo e, para a raquete, utilizamos um retângulo. Os atores podem ou não ter a mesma cor, isso é opcional, ou seja, cada ator pode ter sua singularidade. Entretanto, o que ambos os atores devem ter em comum?

**Resposta:** Ambos os atores devem estar centralizados para que as coordenadas funcionem corretamente quando a lógica de programação for realizada.

Orientação da Alura: A centralização dos objetos que desenhamos fará toda a diferença no projeto devido ao sistema de coordenadas. Se o objeto não estiver centralizado irá acontecer alguns bugs, principalmente quando precisamos inverter a direção do ator.

## ◆ Questão 2 (Scratch)

Para realizar a movimentação da bolinha, utilizamos o bloco mova 10 passos. Esse bloco cria a movimentação de acordo com a direção que o ator estiver apontando. Desse modo, se o ator estiver apontado para a direção de 90 graus, ele vai andar para a direita; se estiver apontado para a direção de 180 graus, o ator vai andar para baixo; se for 0 grau, o ator vai andar para cima; se apontar para -90 graus, o ator vai andar para a esquerda. Se a direção da bolinha for de -180 graus, para qual direção ela irá se deslocar?

**Resposta:** A direção permanecerá a mesma, ou seja, para baixo.

Orientação da Alura: Ao girar 180° no sentido horário ou anti-horário, chegaremos na mesma posição.

### ◆ Questão 3 (Scratch)

A movimentação da raquete é realizada por meio de duas teclas: setas para cima ou para baixo. Essas teclas podem ser alteradas conforme a preferência da pessoa que está programando, ou seja, é possível alterar para as teclas W e S, desde que ela programe de maneira correta. Entretanto, na execução do código abaixo, a movimentação na vertical não está ocorrendo (para cima e para baixo):



Qual é o erro que está ocorrendo neste caso?

**Resposta:** O desenvolvedor utilizou a orientação 'x' para o deslocamento da raquete, algo que não seria viável visto que ela deslizaria horizontalmente. Para o problema ser corrigido de forma adequada, é necessário alterar a orientação para 'y'.

Orientação da Alura: Lembrando que valores positivos a Y deslocam o ator para cima e valores negativos a Y deslocam o ator para baixo.

### ◆ Questão 4 (Scratch)

Com a constante movimentação do ator “bolinha”, as coordenadas x e y também são alteradas. Para que possamos coletar as informações atualizadas da posição do ator “bolinha”, podemos criar uma variável a fim de armazenar os valores das coordenadas. Por meio dessa variável, o ator “raquete do oponente” poderá “ver” as coordenadas x e y do ator “bolinha”. Para essa aplicação, dependendo da sua criação, qual é uma possível limitação da variável?

**Resposta:** A variável deve ser vista por todos os atores.

Orientação da Alura: Uma vez que a variável é “somente para este ator”, é uma variável local. Nesse caso, precisamos de uma variável global que todos os atores possam ver.

## ◆ Questão 5 (Scratch)

Para desenvolver um algoritmo de pontuação do jogo Pong, é necessário avaliar se a bolinha realmente passa sem que a raquete seja rebatida. Para descrever essa situação em código, qual é o principal valor de referência que devemos avaliar para gerar essa pontuação?

**Resposta:** O principal valor a ser avaliado é o 'x' da raquete. Se o valor da coordenada da bolinha for maior que 'x', então significa que ela ultrapassou a raquete.

Orientação da Alura: Lembre-se sempre que, com coordenadas negativas, a referência muda, por essa razão, precisamos alterar as condições lógicas de maior (>) para valores de posição X maior que zero (no caso maior que 229), e menor (<) para valores de posição X menor do que zero (no caso -229).

## ◆ Questão 6 (Scratch)

Para movimentar a raquete do oponente de forma inteligente, uma pessoa desenvolveu o seguinte código no ator bolinha:



E ator raquete oponente:



Após jogar algum tempo, percebeu que o jogo estava muito fácil, pois o oponente errava muito. Com base no código usado pela pessoa, o que ela pode fazer para a raquete do oponente acertar mais a bolinha?

**Resposta:** Ela pode diminuir o valor que está subtraindo a posição y em relação a bolinha.

Orientação da Alura: Quanto mais próximo de 0 for o subtraindo, mais perfeito será o movimento da raquete do oponente. Conforme diminuirmos esse valor, também reduzimos a chance de erro da raquete do oponente.

## ◆ Questão 7 (Editor P5 + JavaScript)

Nesta aula, aprendemos a trabalhar em um editor de texto chamado p5.js. Se quisermos alterar a cor do preview, qual função devemos usar?

**Resposta:** A função `background()`;

Orientação da Alura: Ao modificarmos os valores dessa função, teremos uma alteração na tonalidade do preview.

## ◆ Questão 8 (JavaScript)

Criamos 2 variáveis para movimentar os eixos x e y da bolinha, como ilustra o código abaixo:

```
let velocidadeXBolinha = 6
let velocidadeYBolinha = 6
```

O que acontecerá se alterarmos os valores numéricos para 3, ao invés de 6, como mostra o código abaixo?

```
let velocidadeXBolinha = 3
let velocidadeYBolinha = 3
```

**Resposta:** A velocidade de deslocamento da bolinha será reduzida.

Orientação da Alura: Quanto maiores os valores das variáveis de velocidade para os eixos x e y da bolinha, maior será sua velocidade. Logo, ao diminuirmos esses valores, a velocidade também diminuirá.

## ◆ Questão 9 (JavaScript)

Usamos as condicionais para realizar uma verificação de algum aspecto do código e executar uma ação. Qual é a sintaxe correta para escrevermos uma condicional, em programação na linguagem JavaScript, a fim de verificar se a posição da bolinha no eixo x é maior que a largura da tela?

**Resposta:**

```
if (xBolinha > width){ }
```

## ◆ Questão 10 (JavaScript)

Durante a aula aprendemos como fazer comentários (ou anotações) no nosso código. Se escrevermos uma parte do nosso código da seguinte maneira, o que acontecerá com a segunda linha ao executarmos?

```
xBolinha += velocidadeXBolinha;  
//yBolinha += velocidadeYBolinha;
```

**Resposta:** A linha 'yBolinha' não será considerada na construção do algoritmo.

Orientação da Alura: A bolinha irá se mover apenas no eixo X, o eixo Y não será lido durante a execução, uma vez que está com um comentário.

## ◆ Questão 11 (Editor P5 + JavaScript)

Sem alterar o comportamento do jogo, movemos alguns códigos da função draw para funções que criamos:

- function mostraBolinha()
- function movimentaBolinha()
- function verificaColisaoBorda()

Sabendo disso, podemos definir que função é:

**Resposta:** É um conjunto de código que realiza uma tarefa específica.

Orientação da Alura: Cada função criada é responsável por um comportamento, tornando a leitura do código mais intuitiva e simples.

## ◆ Questão 12 (Editor P5 + JavaScript)

Meu objetivo agora é desenhar uma raquete na tela com as funções pré-existentes na biblioteca do p5.js. O que acontece se eu apenas chamar uma função raquete() no código? **Respostas:** Aparecerá uma mensagem de erro acusando ausência de nome.

Orientação da Alura: Nossa raquete será um retângulo. No p5, já existe uma função predefinida para desenhá-lo, esta chamada de rect(). É possível criar uma função chamada raquete() e obter o mesmo resultado na tela. Mas para isso, devemos inserir a função rect() dentro da função raquete() e chamá-la na função draw(). Para facilitar, vamos usar somente a função rect() e criar nossa raquete através dela.

## ◆ Questão 13 (JavaScript)

Na função para verificar a colisão com a raquete, criamos a seguinte condicional:

```
function verificaColisaoRaquete(){  
  if (xBolinha - raio < xRaquete + raqueteComprimento  
    && yBolinha - raio < yRaquete + raqueteAltura  
    && yBolinha + raio > yRaquete){  
    velocidadeXBolinha *= -1;  
  }  
}
```

Ao utilizarmos a sintaxe &&, estamos nos referindo a qual operador lógico em JavaScript?

**Resposta:** AND, ou seja, a instrução só será executada caso todas as condições sejam satisfeitas.

Orientação da Alura: Com o operador AND (e em inglês), a condicional SE só será executada quando todas as operações matemáticas forem verdadeiras.

## ◆ Questão 14 (GitHub)

Criamos o código que verifica a colisão da bolinha com a nossa raquete verificando a posição x e y de cada um deles. Porém, acessando a documentação do P5 Js, vimos que existe uma solução disponível no GitHub, para reconhecer a colisão entre objetos 2D. Sabendo disso, analise as afirmações abaixo e marque as verdadeiras.

**Respostas:**

- É possível compartilhar um código com a solução desenvolvida por nós para que outras pessoas possam usar.
- O GitHub permite a socialização com pessoas que possuem os mesmos interesses que você.

## ◆ Questão 15 (JavaScript)

Durante a construção da raquete do oponente, percebemos que não precisaríamos criar a função mostraRaqueteOponente(), pois a função mostraRaquete() já continha os parâmetros necessários. Selecione a alternativa que explica por que decidimos usar a função existente em nosso código em vez de criar uma nova.

**Resposta:** Pois assim usamos menos linhas de código, já que nossos parâmetros são muito semelhantes.

## ◆ Questão 16 (JavaScript)

Até o momento no jogo Pong, podemos visualizar o código abaixo dentro da função draw:

```
function draw() {  
  background(0);  
  mostraBolinha();  
  movimentaBolinha();  
  verificaColisaoBorda();  
  mostraRaquete(xRaquete, yRaquete);  
  movimentaMinhaRaquete();  
  verificaColisaoRaquete(xRaquete, yRaquete);  
  mostraRaquete(xRaqueteOponente, yRaqueteOponente);  
  movimentaRaqueteOponente();  
  verificaColisaoRaquete(xRaqueteOponente, yRaqueteOponente);  
}
```

Sabendo disso, analise as alternativas abaixo e marque aquela que mostra a função correta que está relacionada à colisão da raquete do oponente.

**Resposta:**

```
verificaColisaoRaquete(xRaqueteOponente, yRaqueteOponente)
```

## ◆ Questão 17 (JavaScript)

Um aluno ingressou na plataforma da Alura e foi muito bem na construção de seu jogo com Scratch, mas ainda está meio perdido durante a escrita e leitura de algumas linhas de códigos do jogo com a linguagem JavaScript. Em especial com as barras // que aparecem antes de algumas frases ou blocos de códigos.

```
// variáveis da raquete  
let xRaquete = 5;  
let yRaquete = 150;  
let comprimento = 10;  
let altura = 90;  
  
// variáveis da raquete oponente  
let xRaqueteOponente;  
let yRaqueteOponente ;
```

O que será executado quando as linhas //variáveis da raquete e //variáveis da raquete oponente do oponente forem lidas?

**Resposta:** Nada será executado, pois essas linhas são apenas comentários.

Orientação da Alura: Todo código que possua // em seu início, não será executado pelo Javascript.

## ◆ Questão 18 (JavaScript)

Durante o curso, após movimentar a raquete do oponente de forma automática, suponha que um aluno ou uma aluna resolveu criar um placar do jogo, mostrando os pontos ganhos por cada raquete. Para isso, desenvolveu a seguinte função:

```
function marcaPonto(){  
  if (xBolinha > 590){  
    pontosDoOponente += 1;  
  }  
  if (xBolinha < 10){  
    meusPontos += 1;  
  }  
}
```

Porém, ao executar o programa, percebeu que algo não funcionou como esperado. Com base no código, podemos afirmar que:

**Resposta:** Quando eu fizer um ponto, este ponto será atribuído ao oponente.

## ◆ Questão 19 (JavaScript)

Para deixar o jogo mais emocionante e criarmos uma atmosfera de ação para ele, escolhemos colocar uma trilha sonora, que é executada quando o jogo é iniciado. Dentro de um jogo, são necessárias algumas ações para reproduzir um áudio ou uma trilha em formato de loop (repetidamente). Selecione as alternativas que apresentam partes necessárias nesse processo:

**Respostas:**

- Carregar o áudio com a trilha que queremos tocar.
- Criar a função `preload()` e atribuir a variável o resultado da função `loadSound()` com o caminho da trilha e sua extensão.
- Criar uma variável para armazenar a trilha em tempo de execução do nosso jogo.