

# Raquete do jogador

## Unidade

### Lógica de programação: desenvolvendo um jogo estilo Pong

## Questão 1 – Criando uma raquete

Meu objetivo agora é desenhar uma raquete na tela com as funções preexistentes na biblioteca do p5.js. O que acontece se eu apenas chamar uma função **raquete()** no código?

- ☐ a) Uma nova biblioteca será importada para o p5.js.
- ☐ b) Será criada uma raquete, que será chamada na função **draw()**.
- ☐ c) A função será automaticamente definida.
- ☐ d) Uma mensagem de erro aparecerá, avisando que o nome não está definido.

**Alternativa A**, incorreta. Para importar uma biblioteca, é preciso realizar o download ou consumir de um repositório on-line, como o GitHub. Além disso, também é possível construir sua própria biblioteca.

**Alternativa B**, incorreta. É importante lembrar que, ao chamar uma função em *draw()*, é preciso escrevê-la no nosso código; caso contrário, ela não será localizada. Lembre-se de como fizemos na função *mostraBolinha()*.

**Alternativa C**, incorreta. Para consumir uma função preexistente, precisamos inserir as informações e comportamento desejado no código ou podemos consultar tanto as funções preexistentes na biblioteca do p5.js quanto em uma outra biblioteca disponível.

**Alternativa D**, correta. 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 2 – Criando a raquete do jogador com a função `rect`

No jogo Pong, usamos um retângulo para fazer a raquete do jogador. No p5.js, usamos a função **`rect`** com quatro números para desenhar esse retângulo. Cada parâmetro diz algo diferente sobre o retângulo: onde ele começa na tela e o seu tamanho. E devem ser declarados obedecendo uma determinada sequência.

Ordene **todos os blocos** para colocar na ordem certa os parâmetros que você precisa usar na função **`rect`** para desenhar a raquete:

altura

coordenada x

rect

coordenada y

largura

Escreva a sequência correta de blocos nas linhas a seguir:

**Sequência correta:** rect | coordenada x | coordenada y | largura | altura

**Comentário:** a função `rect` possui os parâmetros correspondentes à posição horizontal e à posição vertical. Logo em seguida, os parâmetros que correspondem ao tamanho da raquete devem ser definidos.

## Questão 3 – Implementando a movimentação da raquete

Após criarmos a bolinha do jogo Pong, foi a vez da raquete. Começamos criando uma raquete que será movimentada pelo teclado. Em seguida, criamos uma função **`movimentaRaquete`** que, para ser executada, foi chamada na função **`draw`**. Ordene **todos os blocos** de modo que a função **`movimentaRaquete`** seja declarada corretamente dentro de **`draw`**.

`movimentaRaquete();`

`draw(){`

`function`

Escreva a sequência correta de blocos nas linhas a seguir:

**Sequência correta:** function | draw(){ | movimentaRaquete();}

**Comentário:** a palavra *function* é utilizada apenas na criação da função. Para chamar a função *movimentaRaquete* em *draw*, escrevemos o nome da função, que é *minhaRaquete* e, então, acrescentamos os parênteses e adicionamos o ponto e vírgula, que finaliza a instrução na programação.