

## Implementando a remoção

Joana fez o seguinte código para implementar a funcionalidade de remoção de livros:

```
<!-- lista.marko -->

<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1> Listagem de livros </h1>

    <table id="livros">
      <tr>
        <td>ID</td>
        <td>Título</td>
        <td>Preço</td>
        <td>Editar</td>
        <td>Remover</td>
      </tr>
      <tr id="livro_${livro.id}" for (livro in data.livros)>
        <td>${livro.id}</td>
        <td>${livro.titulo}</td>
        <td>${livro.preco}</td>
        <td><a href="#">Editar</a></td>
        <td><a href="#" data-ref="${livro.id}" data-type="remocao">Remover</a></td>
      </tr>
    </table>

    <script src="/estatico/js/remove-livro.js">
    </script>
  </body>
</html>
```

```
// rotas.js

module.exports = (app) => {

  // demais rotas.

  app.delete('/livros/:id', function(req, resp) {
    const id = req.params.id;

    const livroDao = new LivroDao(db);
    livroDao.remove(id)
      .then(() => resp.status(200).end())
      .catch(erro => console.log(erro));
  });
};
```

Sobre o código de nossa colega, e considerando que, assim como na aula, o arquivo `remove-livro.js` está na pasta `src/app/public/js` e implementado da forma correta, marque as alternativas que indicam afirmativas corretas.

Selecione uma alternativa

- A** Não irá funcionar como esperado, pois a forma correta de obter o ID do livro passado na URL seria `req.id`.
- B** O código não está completo, pois faltou configurar o *middleware* do Express para acesso a arquivos estáticos do projeto.
- C** Não irá funcionar como esperado, pois a forma correta de devolver o *status* da resposta seria fazendo `resp.end(200)`.