

Implementando o cadastro

Cristina fez o seguinte código para o cadastro de livros:

```
// rotas.js

const LivroDao = require('../infra/livro-dao');
const db = require('../config/database');

module.exports = (app) => {

  // demais rotas criadas.

  app.get('/livros/form', function(req, resp) {
    resp.marko(require('../views/livros/form/form.marko'));
  });

  app.post('/livros', function(req, resp) {

    const livroDao = new LivroDao(db);

    livroDao.adiciona(req.body)
      .then(resp.redirect('/livros'))
      .catch(erro => console.log(erro));

  });
};

<!-- form.marko -->

<html>
  <body>
    <h1>Cadastro de livros</h1>

    <form action="/livros" method="post">

      <input type="hidden" name="id" />

      <div>
        <label for="titulo">Titulo:</label>
        <input type="text" id="titulo" name="titulo" />
      </div>

      <div>
        <label for="preco">Preço:</label>
        <input type="text" id="preco" name="preco" placeholder="150.25" />
      </div>

      <div>
        <label for="descricao">Descrição:</label>
        <textarea cols="20" rows="10" id="descricao" name="descricao" placeholder="Descrição do livro" />
      </div>

      <input type="submit" value="Salvar"/>
    </form>
  </body>
</html>
```

```
</form>
</body>
</html>
```

```
// livro-dao.js

class LivroDao {

  constructor(db) {
    this._db = db;
  }

  adiciona({titulo, preco, descricao}) {
    return new Promise((resolve, reject) => {
      this._db.run(`
        INSERT INTO livros (
          titulo,
          preco,
          descricao
        ) values (?, ?, ?)
      `,
      [
        titulo,
        preco,
        descricao
      ],
      erro => {
        if (erro) {
          return reject('Não foi possível adicionar o livro!');
        }

        resolve();
      });
    });
  }

  // demais métodos do dao.
}

module.exports = LivroDao;
```

No entanto, nossa colega não tem certeza se o código dela está correto e se os dados do livro serão efetivamente inseridos no banco de dados.

Sobre isso, podemos afirmar como sendo correto:

Selecione uma alternativa

A O código irá funcionar corretamente, inserindo os dados do livro no banco de dados como esperado.

B O código não irá funcionar, uma vez que o método `adiciona()` da classe `LivroDao` deveria receber um livro quando no código de nossa colega recebe de maneira incorreta `{titulo, preco,`

```
descricao} .
```

C

As informações não serão cadastradas como esperado, pois é necessário utilizar um middleware para obter as informações enviadas no corpo da requisição `POST` .