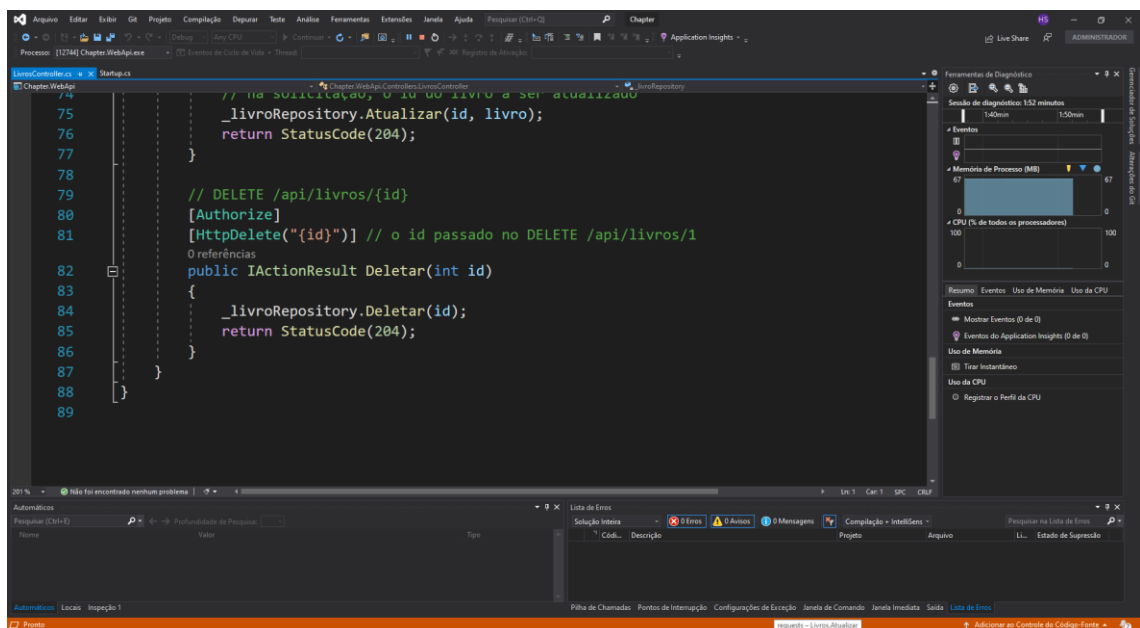


## Cuidando dos erros da sua aplicação

Dentro do contexto de LivrosController, nós criamos as rotas de /api/livros. Cada método é responsável por um verbo.

Para toda aplicação que você desenvolver, existe um cenário que sempre deve ser levado em consideração: tratar os erros e exceções da sua aplicação.

Imagem representando o controller de livros com a ação de deletar.



## Banco de dados

Imagem representando a solicitação para deletar um livro que não existe com este identificador.

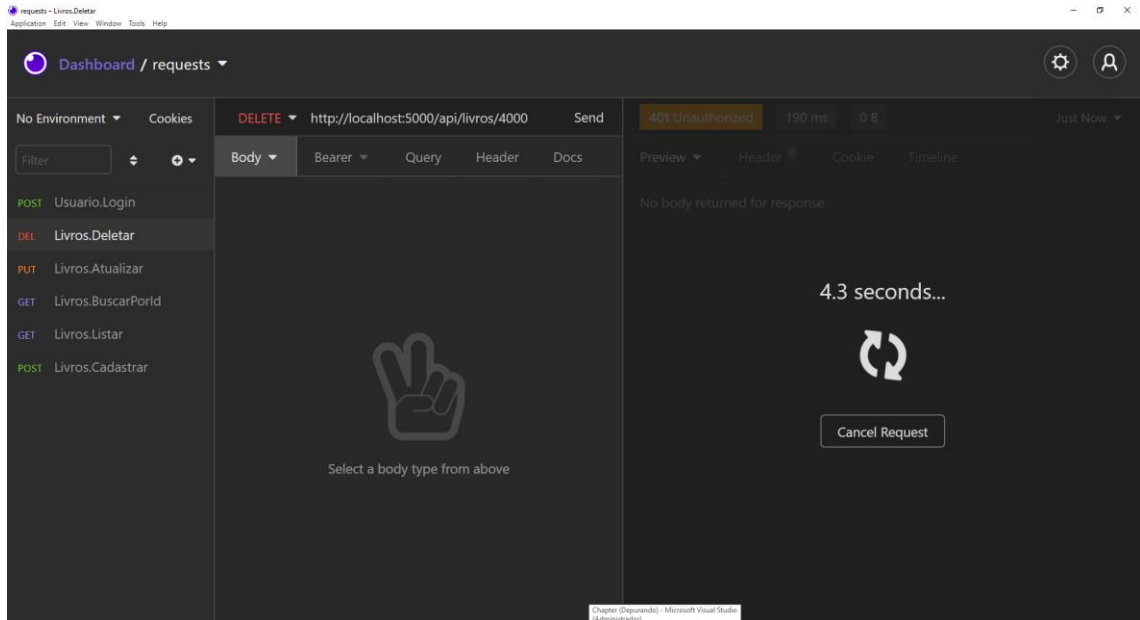
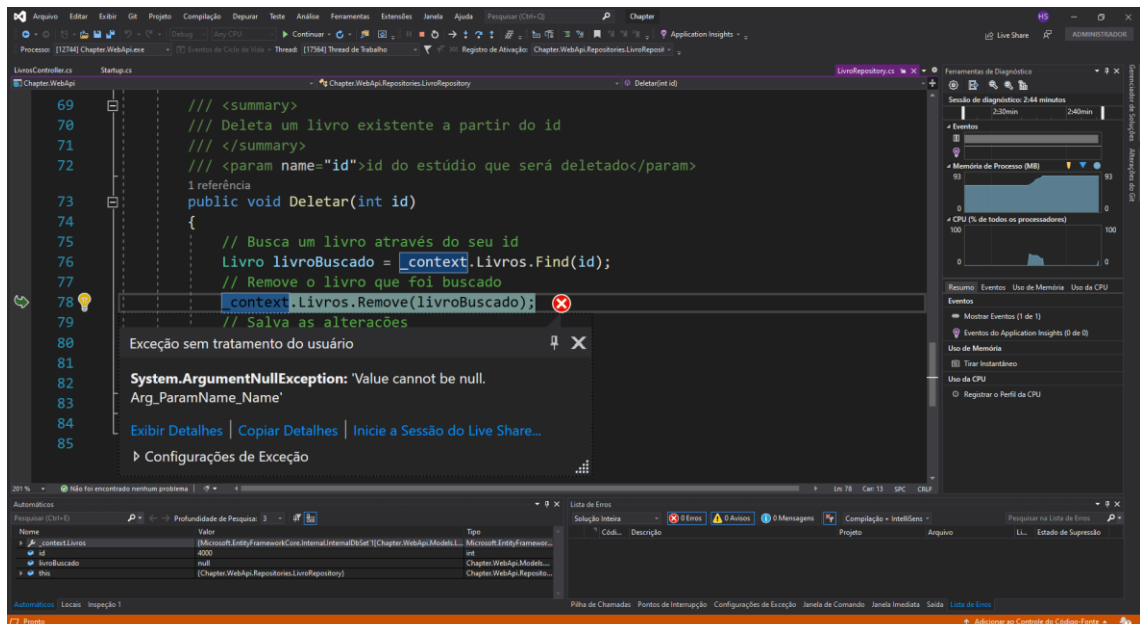


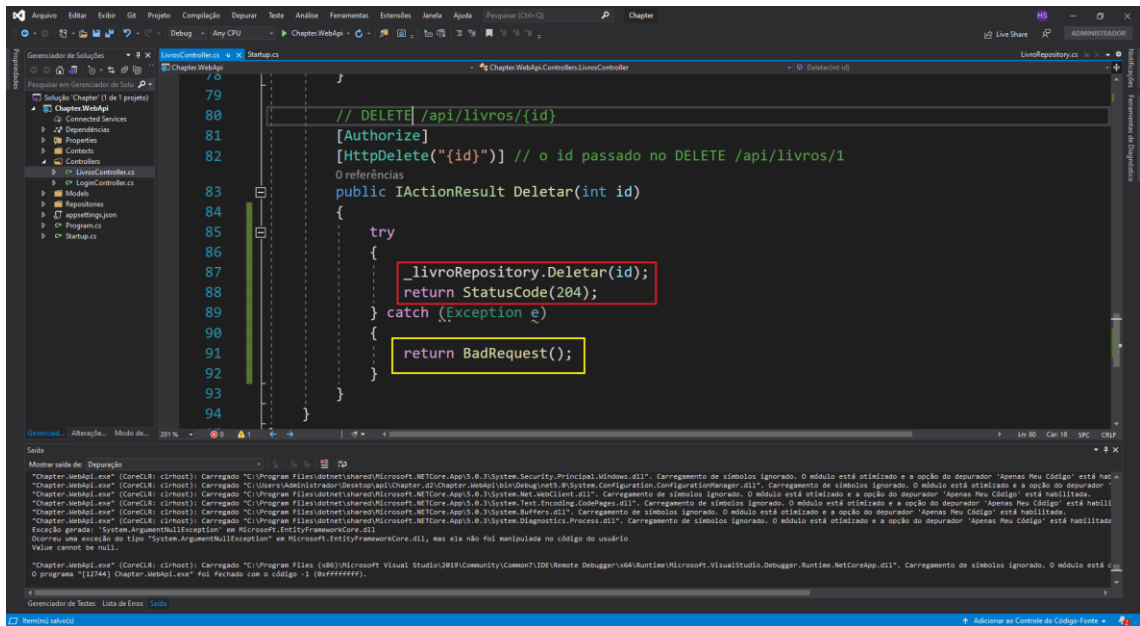
Imagem apresentando um erro na aplicação e parando a execução.



## Banco de dados

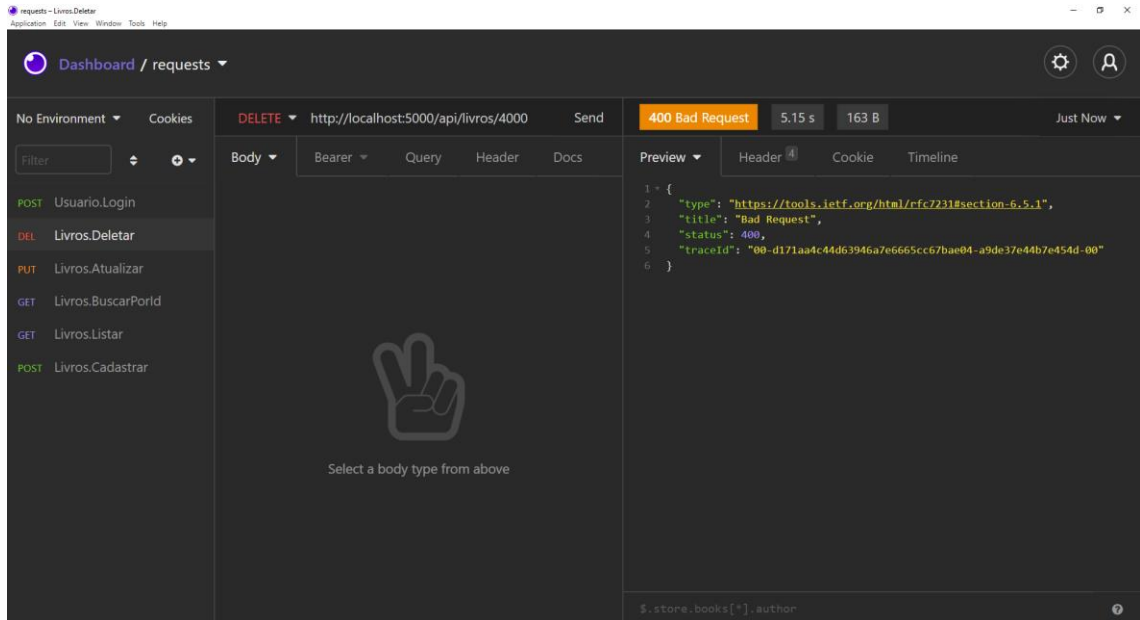
Nesses cenários de exceção e tratamentos de erros em que um bloco de código pode não ser atendido, utilizamos o bloco try/catch.

Imagem para representar a tentativa de deletar um livro. Caso algum erro ocorra, uma mensagem de BadRequest será disparada para o cliente.



```
// DELETE /api/livros/{id}
[Authorize]
[HttpDelete("{id}")] // o id passado no DELETE /api/livros/1
public IActionResult Deletar(int id)
{
    try
    {
        _livroRepository.Deletar(id);
        return StatusCode(204);
    }
    catch (Exception e)
    {
        return BadRequest();
    }
}
```

Imagem apresentando o status code 400 ao tentar deletar um livro que não existe.



Dessa maneira, a execução da aplicação não será interrompida e uma mensagem irá informar ao cliente, o status da solicitação.