

03

Fixando o conhecimento

Considere a classe `LojaApiClient`, responsável pelos métodos de consumo de uma api de uma loja virtual.

Considere também o controlador `ProdutoController` que possui actions para produtos. Veja que no código do controlador ainda utilizamos o repositório de produtos.

```
public class ProdutoController: Controller
{
    private readonly IRepository<Produto> _repo;

    public ProdutoController(IRepository<Produto> repo)
    {
        _repo = repo;
    }

    public IActionResult Index()
    {
        return View(_repo.Todos);
    }

    public IActionResult Excluir(int id)
    {
        var produto = _repo.BuscarPorId(id);
        if (produto != null)
        {
            _repo.Remover(produto);
            return RedirectToAction("Index");
        }
        return NotFound();
    }
}
```

Que mudanças são necessárias para que `ProdutoController` passe a consumir `LojaApiClient`?

Seleciona uma alternativa

- A** Trocar o código que usa o repositório por um que consuma a API, por exemplo no método Index:

```
public IActionResult Index()
{
    var lista = _api.GetProdutosAsync();
    return View(lista);
}
```

- B** solicitar uma instância de `LojaApiClient` via construtor do controlador
substituir o código que usa o repositório por chamadas aos métodos da classe cliente
substituir as assinaturas das actions para deixá-las assíncronas com o uso das palavras reservadas
async/await quando necessário e retornando uma `Task<ActionResult>`

