

Desafio #4.1

Sistema de Controle de Bicicletário - Verificação

Descrição

Você e sua equipe acabam de receber a documentação dos requisitos de software (leia [aqui](#) e [aqui](#)) para um sistema de controle de bicicletas de uma empresa que faz o aluguel em totens disponibilizados na rua.

Além disso, toda a API já foi previamente modelada e documentada utilizando o Swagger. Foi dividida em três microsserviços denominados Ciclista, Equipamento e Serviços Externos. Acesse [aqui](#).

A missão da sua equipe, antes de implementar esta API, é revisar o material para verificar se ela de fato contempla a especificação dos requisitos e evitar futuros problemas.

Instruções

- Data limite de entrega: 01/04/2024 (quarta-feira).
- Prepare um relatório em texto simples que mostre quais *endpoints* atendem os casos de uso.
- Importante 1: esta API representa o *back-end* da aplicação, então você deve imaginar que existirá um *front-end* e, em conjunto, ambos (*back* e *front*) formam o sistema. De toda forma, você só implementará o *back-end*.
- Importante 2: os três microsserviços (Ciclista, Equipamento e Serviços Externos) possuem integração entre si. Esta informação não está detalhada no Swagger, você deve detalhar isto no seu relatório.
- Utilize o esquema similar ao exemplificado abaixo em um arquivo TXT:

UC03 –

Alugar bicicleta (fluxo principal)

Passo X*:

[FRONT]** Operação 1 (Microsserviço A)

[BACK]** Operação 1 [R1]*** (Microsserviço B)

Passo Y:

[FRONT] Operação 2 [R2] (Microsserviço A)

UC03 – Alugar bicicleta (fluxo de exceção 3)

Passo Z:

[FRONT] Operação 3 (Microsserviço A)

UC06 – Alterar dados do ciclista

Passo W:

[FRONT] Operação 1 (Microsserviço C)

...

* Representa um passo do caso de uso em questão. Note que diferentes passos podem invocar diferentes operações, imaginando que partes dos casos de uso serão tratadas em diferentes momentos ou telas do front-end.

**[FRONT] indica que o endpoint/operação será invocado pelo front-end da aplicação. O front-end não será implementado na disciplina.

[BACK] indica que o endpoint/operação será invocado pelo endpoint/operação imediatamente anterior. **IMPORTANTE: O endpoint/operação só invocará outro caso esteja em outro microserviço. Caso contrário, a lógica de negócio estará no mesmo módulo/microserviço e, portanto, é uma chamada de função interna que não precisa ser descrita neste mapeamento.

***[RN] - indica que a regra de negócio N é tratada na respectiva operação.

O exemplo acima (para o fluxo principal) deve ser lido/interpretado da seguinte maneira: "para executar o fluxo principal do UC3, no Passo X o front-end invoca a operação 1 no microserviço A que então invoca a operação 2 do microserviço B devido à R1. Em seguida, no passo Y, o front-end invoca a operação 2 no microserviço A devido à R2".