

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: 24/08/2023 (quinta-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:

UC01 – Cadastrar Ciclista
 Microsserviço A
 [FRONT]* Operação 1
 [BACK]* Operação 2 [R1]**
 Microsserviço B
 [FRONT] Operação 1 [R2]
UC02 – Confirmar email
 Microsserviço E
 [FRONT] Operação 4
...

*[FRONT] indica que o endpoint será invocado pelo front-end da aplicação.

*[BACK] indica que o endpoint será invocado pelo endpoint imediatamente anterior.

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