

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 microserviços 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

- Prepare um relatório em texto simples que mostre quais endpoints atendem os casos de uso.
- Importante: 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.
- Importante²: os três microserviços (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
Microserviço A
 [FRONT]* Operação 1
 [BACK]* Operação 2 [R1]**
Microserviço B
 [FRONT] Operação 1 [R2]
UC02 – Confirmar email
Microserviço 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.

Informações Adicionais

Para complementar o plano de estudos de vocês, material abaixo pode ser utilizado para conhecer o essencial de Swagger:

- Curso - Como usar o Swagger
<https://youtube.com/playlist?list=PLx6gdu4s3nkcQ4qIUYJlXY4saCfpg4XAi>