

Aula 08/05/2023

- 1) Teste de integração: requisição para API
- 2) Teste de carga: teste normal de get dentro de um for
- 3) Teste de processo: teste de uma linha de produção separando as etapas por 'it'

1. Usuário - Camilly

- ☒ ~~Fazer login com o usuário~~
- ☒ ~~Pegar todos os usuários do sistema 200 vezes~~
- ☒ ~~Fazer login, Alterar foto de perfil, Ver Gerente do seu departamento~~

2. Demanda - João

- ☒ ~~Pegar todas as demandas de uma pessoa~~
- ☒ ~~Pegar todas as demandas do banco de dados 100 vezes~~
- ☒ ~~Cadastrar, Classificar e Adicionar informações~~

3. Proposta - Camilly

- ☒ ~~Pegar todas as propostas que não estão em uma pauta~~
- ☒ ~~Pegar uma mesma propostas 200 vezes~~
- ☐ Cadastrar Proposta, Editar centros de custo

4. Pauta - Diego

- ☒ ~~Pegar todas as pautas do sistemas~~
- ☒ ~~Pegar todos os arquivos das pautas do sistema 50 vezes~~
- ☐ Cadastrar Pauta, Adicionar parecer comissão

5. Ata - Diego

- ☒ ~~Pegar todas as atas do sistemas~~
- ☒ ~~Pegar todas as atas do sistemas 100 vezes~~
- ☐ Cadastrar Ata, Adicionar parecer da Direção Geral

```
const url = "http://localhost:8443/sod"
```

```
const pessoaLogin = {
  senha: 123,
  email: "romario@gmail.com"
}

let headers = {
  'Cookie': ""
}
```

```

    it('pegar token', () => {
      cy.request("POST", url + "/login/auth/cookie",
        pessoaLogin).as("TodoRequest")
      cy.get("@TodoRequest").then((response) => {
        headers['Cookie'] = "jwt=" + response.body.value
        cy.setCookie("jwt", headers['Cookie'])
      })
    })

    it("pegar demanda", () => {
      cy.request({
        method: 'GET',
        url: urlDemanda,
        headers
      }).then(response => {
        console.log(response);
      })
    })
  })

```

```

Cypress.Commands.add('fileRequest', (filePath, requestOptions) => {
  return cy
    .fixture(filePath, 'binary')
    .then(file => {
      const blob = Cypress.Blob.binaryStringToBlob(file);
      const formData = new FormData();
      formData.append("demanda",
        JSON.stringify(demandaCadastrar))
      formData.set('pdfVersaoHistorico', blob);

      return cy.request({
        ...requestOptions,
        body: formData,
      });
    })
    .then(response => {
      // const dec = new TextDecoder();
      // return JSON.parse(dec.decode(response.body));

      console.log(response);
    });
});

```

```
it("Requisição para criar uma demanda", () => {

    cy.fileRequest('../e2e/Assets/add.png', {
        method: 'POST',
        url: urlDemanda,
        headers
    })

    // formData.append("pdfVersaoHistorico")

    // cy.request(
    //     {
    //         method: 'POST',
    //         url: urlDemanda,
    //         body: formData,
    //         headers
    //     }
    // )
    // .as("TodoRequest")

    // cy.get("@TodoRequest").then((response) => {
    //     console.log(response);
    // })

    })
```