Laboratório - Criação de repositório, definição do projeto e HTML com funcionalidade principal

Pré-requisitos: node.js, express e Javascript.

Neste laboratório, você vai criar o repositório do seu projeto no GitHub, desenvolver um pequeno servidor backend usando Node.js com Express e criar um arquivo HTML básico com a ideia da funcionalidade principal.

Passo 1: Criação de um Repositório no GitHub

A ideia nesse passo é criar um repositório para seu projeto no github (recomenda-se criar um para o backend e outro para o frontend). Atualize o README do seu projeto com as informações pertinentes sobre o seu projeto: uma breve explicação sobre a ideia principal.

Passo 2: Inicializar um projeto com Node.js

No terminal, inicialize um projeto Node.js:

1) *npm init -y*

Com esse comando, será criado um arquivo package.json que irá conter informações importantes sobre o seu projeto.

Dentro do diretório do seu projeto instale o express com - *npm install express* - e crie um arquivo server.js. Abaixo há duas versões para esse arquivo, sem express e com express.

sem express - podemos ver a utilização do protocolo http de forma direta

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, World!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Nesse ponto acabamos de criar um servidor HTTP básico que responde "Hello World" para qualquer requisição feita no endereço http://127.0.0.1:3000.

com express

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello world!');
});

app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}/`);
});
```

Esse código cria um servidor com Express que responde "Hello World" quando alguém acessa a rota raiz (/). É uma versão mais enxuta e moderna do servidor HTTP básico feito com o módulo nativo http. Ambos os códigos funcionam, porém utilizaremos Express na nossa disciplina. Se você está criando um projeto mais simples e quer controle total, o http nativo serve. Mas **para qualquer aplicação minimamente robusta**, **Express é a melhor escolha** por ser mais rápido de escrever, mais fácil de manter e com muito mais funcionalidades prontas.

Adicione ao arquivo package.json criado, na propriedade scripts, a seguinte linha *start*:

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "start": "node server.js"
},
```

Rode seu projeto através do comando “npm run start” e verifique no navegador o endereço “localhost:3000”, o qual seu servidor está hospedado localmente.

Passo 3: Adicionar um Arquivo HTML

1. No diretório do projeto, crie uma pasta chamada public;
2. Dentro da pasta public, crie um arquivo chamado index.html;
3. Adicione uma estrutura de texto, como ensinado na sala, com a descrição sobre seu projeto e as principais funcionalidades que serão implementadas, procure utilizar os elementos de html comentados em sala de aula, para estruturação do arquivo.

Passo 4: Servir o Arquivo HTML

1. Modifique o server.js para servir o arquivo HTML estático:
    a. adicione as linha no server.js

```
const path = require('path');

app.use(express.static(path.join(__dirname, 'public')));
```

Passo 5: Acessar HTML

Rode o projeto novamente e acesse no navegador http://localhost:3000/{index.html}.

Aqui já conseguimos criar uma estrutura servidor/cliente que demonstra um pouco do início do desenvolvimento web.

Perguntas:

➢ Adicione as respostas para as seguintes perguntas no seu arquivo HTML:

1) E se quiséssemos garantir que os objetos req e res fossem sempre utilizados da forma correta, sem depender só da nossa memória ou da sorte? Com base no que discutimos sobre TypeScript, **como ele poderia nos ajudar nesse cenário?**

2) Como você imaginaria esse mesmo código escrito em TypeScript? O que mudaria na estrutura do código? Quais benefícios você esperaria ter? E quais possíveis dificuldades surgiriam?