

FIC 2 – Módulo II

Aula 3 – Leitura de arquivos no Servidor Web

Criação de páginas
customizadas com Node.js

Leonardo Dalvi Pio



INSTITUTO FEDERAL
Espírito Santo

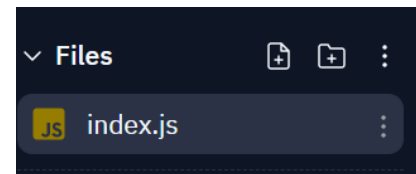
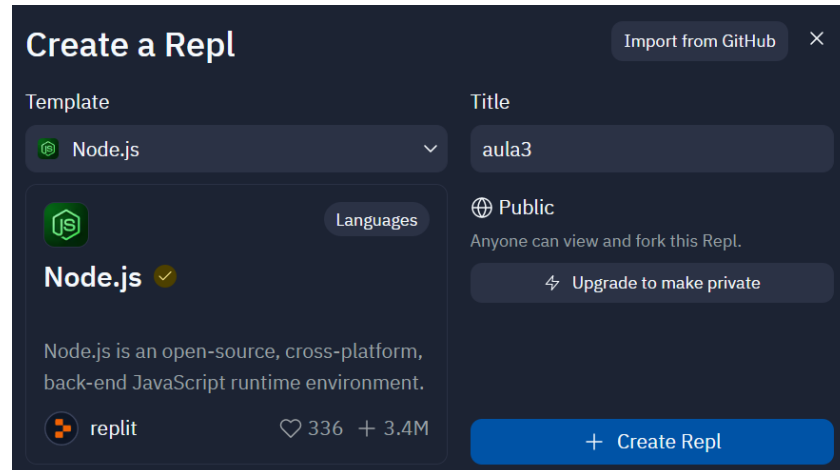
Leitura de arquivos no Servidor Web

- Criação de um servidor web.
- Leitura de um arquivo no servidor web.
- No Replit, se o pacote/recurso (***pagkage.json***, ***express***) não estiver disponível, ele será baixado automaticamente.



Leitura de arquivos no Servidor Web

- No Replit, crie uma pasta nomeada "Aula 3".
- Dentro da pasta, crie um novo arquivo (New Repl). Selecione o Template 'Node.js' e atribua o título como: "aula3". Será incluído o arquivo 'index.js' em '*Files*'.



Leitura de arquivos no Servidor Web

- Dentro da pasta, crie um novo arquivo (New Repl), no formato HTML, com o nome 'site.html'.
- Utilize uma estrutura básica do HTML, conforme código abaixo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Manipulacao de arquivos</title>
  </head>
  <body>
    <h1>Manipulacao de arquivos utilizando o modulo FS</h1>
    <p1>LEITURA</p1>
  </body>
</html>
```

Leitura de arquivos no Servidor Web

- No arquivo `'leitura.js'`, criaremos uma constante nomeada `'express'`, conforme código:

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

- Utilizamos o *require* (requisição) para acessar o módulo `'http'` (importação do módulo `'http'`).
- Em seguida criaremos uma aplicação do módulo FS (*File System*) que trabalha com manipulação de arquivos.

```
const fs = require('fs')
```

- Mais informações sobre o módulo FS:
<https://nodejs.org/api/fs.html>

Leitura de arquivos no Servidor Web

- Utilizaremos a porta 443.

```
const porta = 443
```

- Em seguida, com o objeto 'http', utilizaremos a função `createServer()`.
- Utilizaremos a *Arrow Functions*. Passaremos às devidas variáveis: **req (requisição)** e **res (resposta)**, também bastante utilizadas em inglês, sendo *request* e *response*. As *Arrow Functions* são utilizadas por: ``=>``.

```
const servidor = http.createServer((req, res) => {
```

- Obrigatoriamente, devemos utilizar nessa ordem, requisição e resposta.

Leitura de arquivos no Servidor Web

- Na nossa *Arrow Functions* criada, começaremos com o módulo 'fs', utilizando a função 'readFile()' para ler o arquivo HTML criado.

```
fs.readFile('site.html', (err, arquivo) => {
```

- Logo, no comando acima, o 'readFile' do módulo 'fs' irá ler o arquivo 'site.html' e anexar em 'arquivo', e posteriormente, caso aconteça algum erro no retorno do caso e arquivo (leitura), ele será tratado por 'err'.
- Na resposta (**res**), utilizaremos o writeHead(status:Code, reasonPhrase?), com código 200 (conforme Página [http.cat – status ok](http://cat.status)) e cabeçalho 'Content-Type':'text/html'.

```
res.writeHead(200, { 'Content-Type': 'text/html' })
```

Leitura de arquivos no Servidor Web

- Continuando na resposta (**res**), incluiremos o comando `'write()'`, escrevendo o texto lido do arquivo HTML.

```
res.write(arquivo)
```

- Indicamos o final da nossa **resposta** com o comando `end()`.

```
res.end( )
```

- Para finalizar, incluiremos o comando `'.listen()'` e a porta de comunicação (porta 443) com o Servidor, apresentando a mensagem: `'Servidor rodando'` (com o comando: `console.log`).

```
servidor.listen(porta, ( ) => {console.log('Servidor rodando') })
```

- Pronto! Já temos um Servidor Web com leitura de arquivos!

Leitura de arquivos no Servidor Web

Codificação final:

```
1  const http = require('http')
2  const fs = require('fs')
3  const porta = 443
4
5  const servidor = http.createServer((req, res) => {
6    fs.readFile('site.html', (err, arquivo) => {
7      res.writeHead(200, { 'Content-Type': 'text/html' })
8      res.write(arquivo)
9      return res.end()
10    })
11  })
12
13  servidor.listen(porta, () => { console.log('Servidor rodando') })
```

Testes!

- Para testar o código, inicie com o botão 'Run' do Replit.
- Na aba 'Console' é possível observar a mensagem: "Servidor rodando".
- Na aba 'Webview' é possível observar a página inicial do site, do arquivo lido (HTML).
- Altere algo no arquivo HTML, por exemplo:
 - `<p1>LEITURA alterado</p1>`
- Abra uma nova aba do código no *browser*. Atualize a página.
- Observe que a página foi atualizada e não foi necessário reiniciar o servidor, isso porque não realizamos nenhuma modificação no código do servidor.



**INSTITUTO
FEDERAL**
Espírito Santo