

FIC 2 – Módulo II

Aula 11 – Upload de arquivos com o módulo formidable

Criação de páginas
customizadas com Node.js

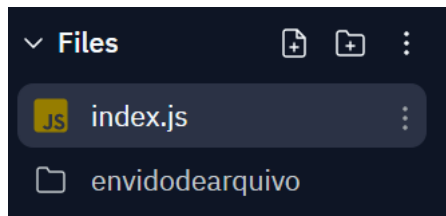
Leonardo Dalvi Pio

Upload de arquivos com o formidable

- Criar um Servidor Web em Node utilizando o módulo *Formidable* para envio de arquivos.
- No Replit, se o pacote/recurso (*pagkage.json*, *express*, *formidable*) não estiver disponível, ele será baixado automaticamente.

Upload de arquivos com o formidable

- No Replit, crie uma pasta nomeada "Aula 11".
- Dentro da pasta, crie um novo arquivo (New Repl).
Selecione o Template 'Node.js' e atribua o nome do arquivo de "index.js" (arquivo do tipo .js).
- Crie uma pasta nomeada: 'envidodearquivo'.



Upload de arquivos com o formidable

- Iniciaremos criando as constantes e importando os módulos. Criaremos o módulo 'http', utilizando a porta padrão = (443 - https) e os módulos 'formidable' e 'fs' com o comando 'require()'.

```
const http = require('http')
const porta = 443
const formidavel = require('formidable')
const fs = require('fs')
```

- Criaremos o servidor:

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

Upload de arquivos com o formidable

- Após criar o servidor, criaremos nossa página principal HTML.

```
res.writeHead(200, { 'Content-Type': 'text/html' })
res.write('<form action = "enviodearquivo" method = "post" enctype = "multipart/form-data">')
res.write('<input type = "file" name = "filetoupload"><br>')
res.write('<input type = "submit" value = "Enviar">')
res.write('</form>')
return res.end()
```

- Na nossa página HTML, criaremos um formulário, onde a função 'action' é o envio de arquivo ('enviodearquivo') com o campo 'file', e o nome desse campo é 'filetoupload'.
- Incluiremos o comando 'listen()' e a mensagem: "Servidor rodando" será exibida.

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


Upload de arquivos com o formidable

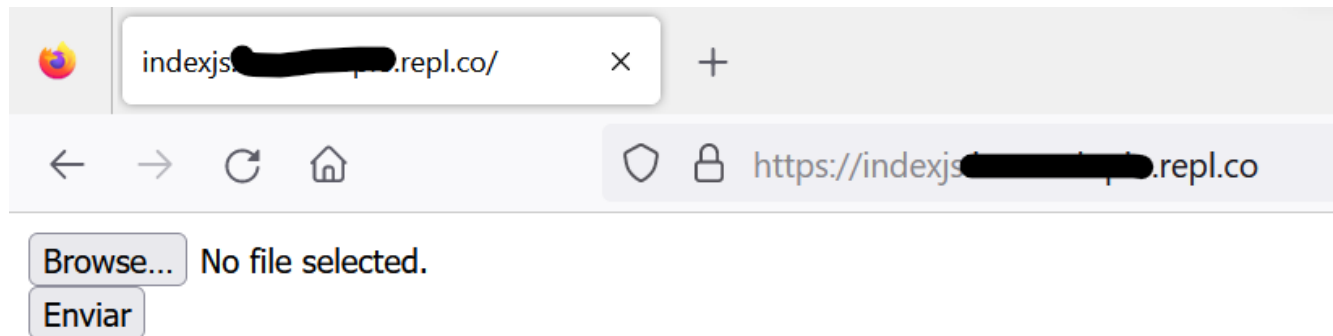
- Código:

```
1  const http = require('http')
2  const porta = 443
3  const formidable = require('formidable')
4  const fs = require('fs')
5
6  const servidor = http.createServer((req, res) => {
7    res.writeHead(200, { 'Content-Type': 'text/html' })
8    res.write('<form action = "enviodearquivo" method = "post" enctype = "multipart/form-data">')
9    res.write('<input type = "file" name = "filetoupload"><br>')
10   res.write('<input type = "submit" value = "Enviar">')
11   res.write('</form>')
12   return res.end()
13 })
14 servidor.listen(porta, () => { console.log('Servidor rodando') })
```



Upload de arquivos com o formidable

- Rodando nosso servidor, abra uma nova aba do código no Browser (ex: <https://indexjs.usuario.repl.co/>). A página abaixo será apresentada:



Upload de arquivos com o formidable

- Em seguida, criaremos um *'if'-else'* dentro da *'const servidor'*, com a rotina de envio de arquivos. Se a requisição (*req.url*) for diferente de *'/enviodearquivo'*, ele irá criar a página *HTML* do nosso servidor. Caso contrário, criaremos a rotina de envio de arquivos.

```
const servidor = http.createServer((req, res) => {  
  if (req.url !== '/enviodearquivo') {  
  
  }  
  else {  
  
  }  
})
```


Upload de arquivos com o formidable

- Começaremos copiando o código da nossa página principal *HTML* e colando dentro de *'if'*.

```
if (req.url !== '/enviodearquivo') {  
  res.writeHead(200, { 'Content-Type': 'text/html' })  
  res.write('<form action = "enviodearquivo" method = "post" enctype = "multipart/form-data">')  
  res.write('<input type = "file" name = "filetoupload"><br>')  
  res.write('<input type = "submit" value = "Enviar">')  
  res.write('</form>')  
  return res.end()  
}
```



Upload de arquivos com o formidable

- Dentro de *'else'*, começaremos criando um *'form'* (uma constante) do tipo *'formidable'* (do módulo *'formidable'*) com o *'IncomingForm()'* que retornará o formulário que foi enviado a ele, conforme abaixo:

```
const form = new formidable.IncomingForm()
```

- Em seguida, devemos passar a nossa requisição (**req**) com uma *Arrow function* tratando: erros, campos e arquivos, conforme código abaixo:

```
form.parse(req, (erro, campos, arquivos) => {
```

Upload de arquivos com o formidable

- Criaremos duas constantes, uma com o URL antigo e outra com um novo. No URL antigo, utilizaremos o mesmo nome empregado no formulário. No nosso caso, foi utilizado o nome `fileupload[0]`, finalizando com o comando `'filepath'`, que seleciona o caminho enviado (caminho do arquivo selecionado).
- O novo URL servirá para apontar o local onde será salvo o arquivo selecionado (Destino) no nosso servidor de arquivos. Utilizaremos o mesmo comando anterior, só que utilizando o comando `' + arquivos.fileupload[0].originalFilename'`, com o nome original do arquivo selecionado.
- Abaixo o código com a criação dos URLs:

```
const urlAntiga = arquivos.fileupload[0].filepath
const urlNova = './enviodearquivo/' + arquivos.fileupload[0].originalFilename
```

Upload de arquivos com o formidable

- Criaremos uma variável `'rawData'` que recebe o método `'fs.readFileSync()'` para ler o arquivo (`urlAntiga`) e retornar seu conteúdo.
- O método `'fs.readFile()'` lê um arquivo de forma assíncrona sem bloqueio. No método `'fs.readFileSync()'`, pode-se ler arquivos de forma síncrona, ou seja, quando o método `'fs.readFileSync()'` é chamado, o programa aguarda até que ele seja executado.

```
var rawData = fs.readFileSync(urlAntiga)
```



Upload de arquivos com o formidable

- Finalizando a parte do envio de arquivos, criaremos uma *Arrow function()* e utilizaremos o módulo *'fs'*. Empregaremos o *'writeFile'* para pegar o URL antigo (origem) e trocar pelo novo URL (destino) e, caso haja um erro, ele será apresentado. O método *'fs.writeFile()'* substitui o conteúdo do arquivo sempre que é chamado. Se o arquivo não existir, ele será criado com o conteúdo que você inserir. Caso o arquivo já exista, o método irá substituir o arquivo criado anteriormente.
- A mensagem "arquivo enviado com sucesso" será exibida e, se acontecer um erro, será emitida uma mensagem (de erro). Encerraremos com o *'res.end()'*, conforme código abaixo:

```
fs.writeFile(urlNova, rawData, function(err) {  
  if (err) console.log(err)  
  res.write("Arquivo enviado com sucesso!")  
  res.end()  
})
```

Upload de arquivos com o formidable

- Codificação final (1/2):

```
1  const http = require('http')
2  const porta = 443
3  const formidable = require('formidable')
4  const fs = require('fs')
5
6  const servidor = http.createServer((req, res) => {
7    if (req.url !== '/enviodearquivo') {
8      res.writeHead(200, { 'Content-Type': 'text/html' })
9      res.write('<form action = "enviodearquivo" method = "post" enctype = "multipart/form-data">')
10     res.write('<input type = "file" name = "filetoupload"><br>')
11     res.write('<input type = "submit" value = "Enviar">')
12     res.write('</form>')
13     return res.end()
14   }
```



Upload de arquivos com o formidable

- Codificação final (2/2):

```
15  else {
16      const form = new formidable.IncomingForm()
17      form.parse(req, (erro, campos, arquivos) => {
18          const urlAntiga = arquivos.fileupload[0].filepath
19          const urlNova = './enviodearquivo/' + arquivos.fileupload[0].originalFilename
20          var rawData = fs.readFileSync(urlAntiga)
21          fs.writeFile(urlNova, rawData, function(err) {
22              if (err) console.log(err)
23              res.write("Arquivo enviado com sucesso!")
24              res.end()
25          })
26      })
27  }
28  })
29  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 os botões de: Escolher arquivo e Enviar.
- Observe dentro da pasta 'enviodearquivo' que não existe nenhum arquivo.
- Selecione o botão 'Escolher arquivo'. Escolha qualquer arquivo e envie.
- Observe que o arquivo foi enviado para dentro da pasta 'enviodearquivo'.



**INSTITUTO
FEDERAL**
Espírito Santo