pulsati

FileSystem e Promise

Require('fs');

Como interagir com os arquivos do computador

FileSystem

Carregar funções para acessar o sistema de arquivos:

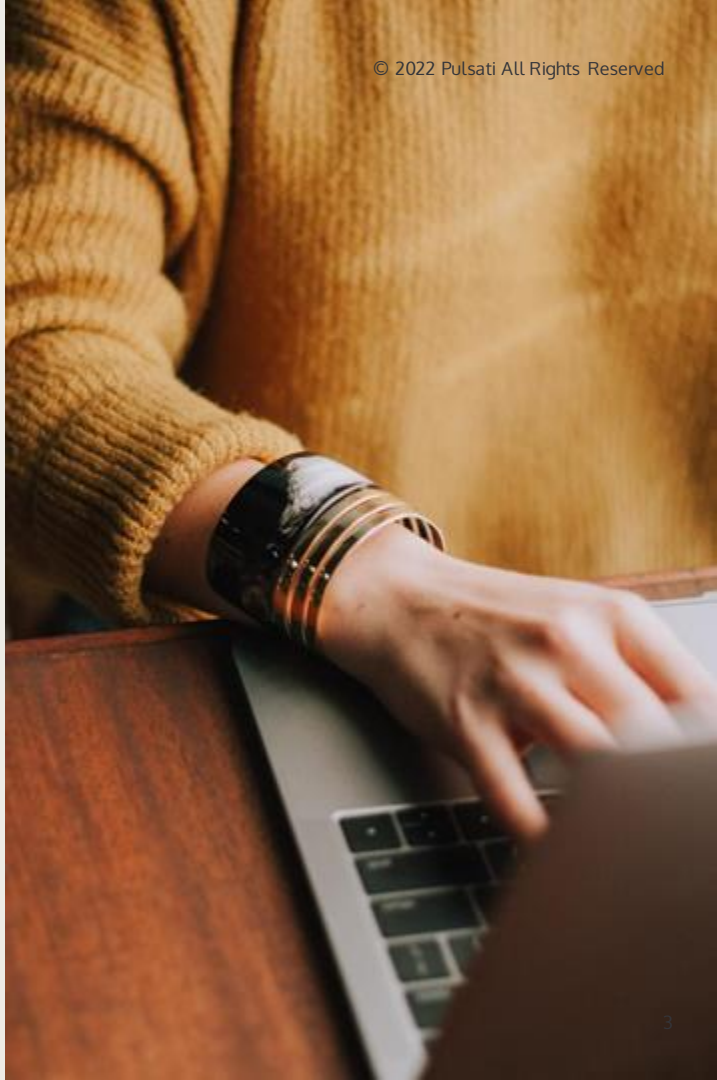
```
$ const fs = require('fs');
```

Escrever em um arquivo:

```
$ fs.writeFile('teste.txt', 'meu texto', (erro) => console.log(erro));
```

Ler um arquivo:

```
$ fs.readFile('teste.txt', 'utf8', (erro, texto) => {
  if (erro) {
    console.log(erro);
  } else {
    console.log(texto);
  }
});
```



Prática

Criar uma classe de Usuario

Criar uma instância de Usuario

Salvar os dados do Usuario em um arquivo
Apresentar no console quando salvou

Carregar os dados de um Usuario
Apresentar no console o resultado

Promise

Implementação assíncrona

O que é uma Promise?

O JavaScript é uma linguagem sequencial, cada linha de código fonte somente é executada depois de finalizar a execução da linha anterior.

O que pode fazer com que nossa aplicação trave totalmente quando houver alguma interação que demore a ser finalizada.

Para esta necessidade, foi implementada a Promise.

Interagindo com o FileSystem usando Promise:

```
exports.salvarTexto = (texto) => {
  const resultado = new Promise((resolve, reject) => {
    fs.writeFile('meu-texto', texto, (error) => {
      if (error) {
        return reject(error);
      } else {
        return resolve()
      }
    });
  });
  return resultado;
}
```

Prática

Continuando a prática anterior

Crie 10 usuários, adicionado na rotina de salvar:
$ setTimeout(() => {}, Math.random(3));

Tente consultar um Usuário, que ainda não tenha salvo

Como garantir que execute somente de salvar?

- Promise.then em sequência
- Promise.all de todas as promises

Exercícios o/

+

Aquecimento

Refatore o exercício de criação da aplicação de gerenciamento de viagens.

Altere para que quando o usuário inserir um aeroporto, voo ou passageiro, essa informação seja salva em um arquivo e retorne para o usuário a confirmação se salvou

Criando uma Aplicação de Livros

Crie uma aplicação de gerenciamento de livros

Funcionalidades:

- A aplicação deve apresentar um menu no console com as opções, cada opção deve ter um número
- O menu de opções deve ter as seguintes opções:
- Adicionar autor, informando: Nome, Código
- Adicionar editora, informando: Nome, Código e Edição
- Adicionar livro, informando: Código do autor, Código do livro
- Listar livros, apresentando as informações: Nome do livro, Edição do livro e Nome do autor
- Buscar livro por nome, solicita pelo terminal, informar o nome do livro
- Buscar livros do autor, solicita pelo terminal, informar o nome do autor
- Quando o usuário digitar o número de uma das opções, a aplicação deve solicitar as informações através do console
- Validar que todas as informações de um autor, editora ou livro sejam preenchidas
- Quando usuário adicionar alguma entidade, esta informação deve ser salva em um arquivo, onde o nome do arquivo seja o código da entidade

Criando outras Aplicações

Conforme a aplicação que foi criada no exercício anterior, pense como poderíamos criar essas outras duas aplicações:

- Crie uma aplicação de gerenciamento de eventos;
- Crie uma aplicação de gerenciamento de filmes.

Obrigado!

pulsati

pulsati.com.br