UTILIZANDO PACOTES DE OUTRA PESSOA:

PROJETO 1: utilizar um pacote que pega uma imagem. Diminuir a imagem
(será utilizado um pacote que consegue redimensionar uma imagem. Assim será necessário apenas chamar as funções que irá realizar as ações)

Considerações iniciais

Parte relacionada as pastas e arquivos que serão necessários para o projeto:

- *criar uma pasta [temp]*
  (para colocar a imagem que já foi redimensionada. Criando assim, uma cópia do original)
- *criar uma pasta [compressed] (para colocar a imagem que já foi comprimida)*
- *criar um arquivo [app.js ou outro nome com .js]*
- *Colocar a imagem a ser trabalhada*
  (podendo ser pedida direto para o usuário caso queira. Basta modificar o codigo)

No início do projeto deve ser criado o pacote desse projeto
(para depois colocar o pacote de comprimir imagem)

No terminal do Vscode ou cmder

- Npm init

Sera realizado algumas perguntas:
- package name: (node) definir Nome
- version: (1.0.0) digita uma versão
- description: digita a descrição
- entry point: (index.js) digita o nome.js
  É usado para definir qual o arquivo principal que unir todos os outros métodos
- test command:
- git repository:
- keywords:
- author:
- license: (ISC)

Confirmar a criação do pacote

Foi criado o pacote
Basta olhar nos arquivos do projeto, terá um arquivo **[package.json]**

Encontra pacotes para o node.js

- *Pesquisar por uma funcionalidade no google, como:* resize image node (redimensiona imagens)
- digitar no terminal o nome do pacote visto no site [https://sharp.pixelplumbing.com/install]

Criar o pacote no meu projeto:

```
npm install sharp
```

Como fica o terminal quando terminou de instalar o pacote

```
C:\Users\me\Desktop\node> npm install sharp
npm WARN saveError ENOENT: no package.json, open 'C:\Users\me\Desktop\node\package.json'
npm WARN enoent ENOENT: no package.json, open 'C:\Users\me\Desktop\node\package.json'
npm WARN node-pre-gyp@0.11.0 No repository field.
npm WARN node-pre-gyp@0.11.0 No license field.

added 8 packages, and audited 8 packages in 3s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Após instalar vai ser criado:

- Uma pasta: [node_modules] onde ficou todos os arquivos do pacote

Para utilizar o pacote no meu projeto deve ver a documentação no site. Onde vai ter como utilizar

Etapas para redimensionar a imagem

(no arquivo app.js)

- *Fazer uma requisição para o pacote instalado*

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

- *Pegar os dados digitados pelo usuário*

```
será usado para receber o nome da imagem a ser trabalhada (recebe string)
let endereco_nomeImagem= process.argv[2];
será usado para receber o valor do comprimento que a imagem deve ficar (recebe números)
let comprimentoImagem= Number(process.argv[3]);
```

- *Criar uma função responsável por pegar os dados digitados no terminal e iniciar o redimensionamento da imagem*

```
function redimencionarImagem (endereco_nomeImagem, comprimentoImagem){
  • Utilizando o pacote de outra pessoa. Que consegue redimensionar imagens
    sharp(endereco_nomeImagem).resize({width:comprimentoImagem}).toFile('./temp/imagemResize.jpg',(error)=>{
        if(error){
          console.log(error);
        }else{
          console.log("imagem redimensionada com sucesso");
        }
    })
}
```

- *Chamar a função*

```
redimencionarImagem(endereco_nomeImagem, comprimentoImagem)
```

OBS: Estrutura para pegar um dado digitado no terminal

```
Let nomeParaVariavel=process.argv[numero];
```

Nome para a variavel a ser utilizada no sistema	process.argv	Número da posição onde essa variavel vai esta quando for colocada no terminal

Obs: escrevendo no terminal

node	nomeProjeto.js	Variavel1	Variavel2
0	1	2	3

OBS: estrutura da função

que é responsável por definir quais os elementos que vão ser trabalhado no pacote de redimensionar imagem

A FUNÇÃO DEVE RECEBER DOIS PARAMETROS. SÃO ELES:
- *a imagem a ser redimensionada (pode ser definida, recebendo a variavel que corresponde a imagem)*
- *O valor do comprimento que a imagem vai ficar (pode ser definida, recebendo a variavel que corresponde ao comprimento desejado)*

```
function redimencionarImagem (Imagem , valorDoComprimento){
    ...........
}
```

```
sharp(endereco_nomeImagem).resize({width:comprimentoImagem}).toFile('./temp/imagemResize.jpg',(error)=>{
})
```

Sharp	chamar a imagem (variavel da imagem)	.resize	• Colocar a teg width e definir um número para o comprimento da imagem ou colocar a variavel que tem o valor do comprimento desejado • Pode ser colocado a altura a imagem também. Basta criar outra variavel que será pedida no terminal ou atribuir diretamente	.toFile	• Colocar o endereço de uma pasta para colocar a imagem quando a imagem tiver sido redimensionada [./temp] • Definir um nome e um formato para a imagem redimensionada [/imagemResize.js]	Função de verificação: mostra o que deu errado ou confirma que a imagem já foi modificada

Chamar a função que redimensiona a imagem. Colocando os elementos a ser trabalhado

```
redimencionarImagem(endereco_nomeImagem, comprimentoImagem)
```