



Curso: Curso rápido Programador Front-End

Professor: Richard Brosler

Unidade Curricular: Lógica de Programação – UC 8

Atividade Prática 1: Codificação de um sistema de contagem de peças

Nessa atividade iremos utilizar os **conteúdos desenvolvidos no desafio 3**.

Para essa atividade trabalharemos em grupos de **2 pessoas** ou **individual**, mantendo distância segura entre os membros.

A proposta dessa atividade consiste em:

A empresa **Savinis**, focada no desenvolvimento de software de alta performance, contratou você para realizar a programação de um sistema de cadastro de peças, que deverá atender os seguintes requisitos:

- Se a peça possuir um peso superior a 100gramas, pode cadastrar.
- Dada a capacidade de cada caixa, caso a lista de peças seja superior a 10, imprima uma mensagem informando não ter capacidade suficiente.
- Caso a peça tenha um nome com quantidade inferior a 3 caracteres, informe uma mensagem de erro.

Para criação desse sistema de cadastro, você, enquanto programador(a) responsável, deverá solucionar os seguintes desafios:

- Aplicar técnicas de programação e ferramentas para desenvolvimento do código.
- Aplicar linguagem de programação JavaScript para implementação do sistema.

Etapas a serem realizadas

- **Etapa 1:** Preparação do diretório da atividade
- **Etapa 2:** Análise do que se pede e elaboração de fluxograma/algoritmo descritivo/pseudocódigo
- **Etapa 3:** Desenvolvimento do código
- **Etapa 4:** Testes
- **Etapa 5:** Subir código para o repositório remoto

Etapa 1

Primeiramente, vamos preparar toda a estrutura que precisamos.

Primeiro, vamos instalar o vscode, que está disponível no endereço: <https://code.visualstudio.com/Download>



Depois instalaremos o NodeJS, que está disponível no endereço: <https://nodejs.org/pt-br/download/>

Na sequência iremos preparar a estrutura dos diretórios para a nossa atividade.

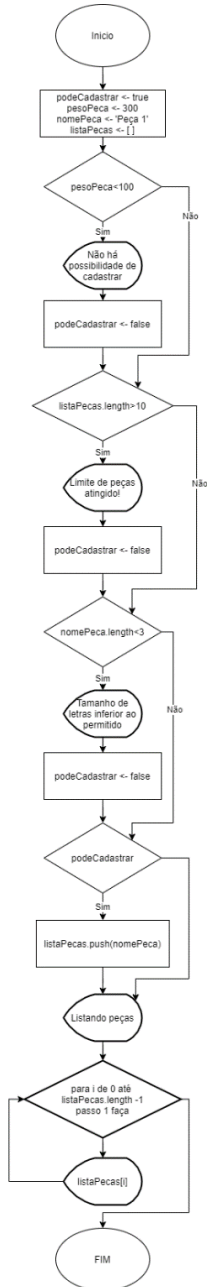
No vídeo abaixo eu explico tudo. Para acessar o vídeo, acesse o endereço: <https://youtu.be/-NI03H3fDa4>

O vídeo está legendado, portanto, podem desligar o áudio para assisti-lo. 😊



Etapa 2

Agora, vamos analisar o que se pede e pensar no fluxograma/algoritmo descritivo/pseudocódigo da atividade.



Algoritmo "pecas"

início

podeCadastrar <- true

pesoPeca <- 300

nomePeca <- 'Peça 1'

listaPecas <- []

se pesoPeca < 100 então

escreva('Não há possibilidade de cadastrar')

podeCadastrar <- false

fim se

se listaPecas.length > 10 então

escreva('Limite de peças atingido!')

podeCadastrar <- false

fim se

se nomePeca.length < 3 então

escreva('Tamanho de letras' +
'inferior ao permitido')

podeCadastrar <- false

fim se

se podeCadastrar então

listaPecas.push(nomePeca)

fim se

escreva('Listando peças')

para i de 0 até listaPecas.length-1 passo 1 faça

escreva(listaPecas[i])

fim para

fim algoritmo

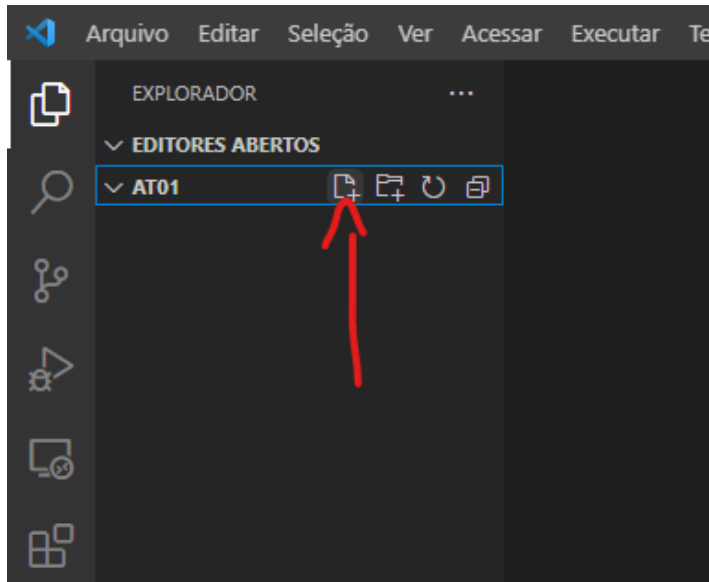
Agora vamos passar o código para Javascript...



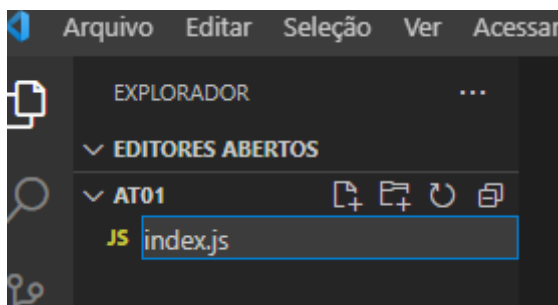
Etapa 3

No VSCode abra a pasta c:\CTProgFrontEnd\seunome\UC8\At01 para iniciar um novo código.

Cliquem no ícone para adicionar novo arquivo como na imagem:



Escrevam index.js como na imagem abaixo:



Depois confirme com a tecla ENTER

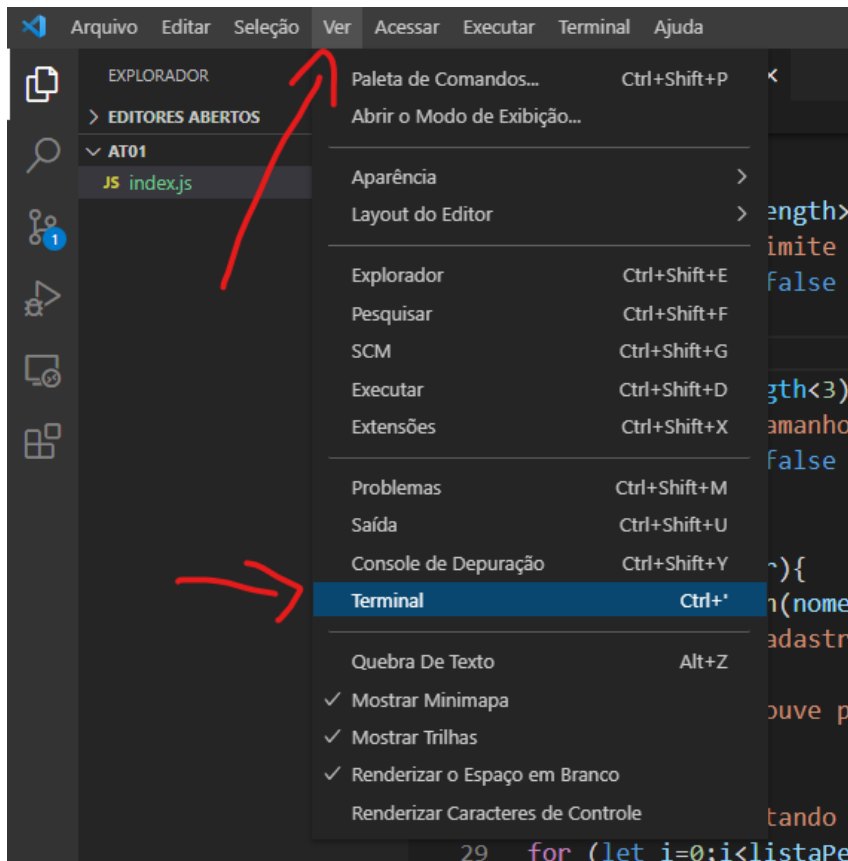
Agora vamos escrever o código abaixo:

```
Bem-vindo(a)  JS index.js U X
JS index.js > [e] pesoPeca
1 let listaPecas=[]
2 let nomePeca='Peça Teste'
3 let pesoPeca=300
4 let podeCadastrar=true
5
6 if (pesoPeca<100){
7   console.log('Não há possibilidade de cadastrar!')
8   podeCadastrar=false
9 }
10
11 if (listaPecas.length>10){
12   console.log('Limite de peças atingido!')
13   podeCadastrar=false
14 }
15
16 if (nomePeca.length<3){
17   console.log('Tamanho de letras inferior ao permitido')
18   podeCadastrar=false
19 }
20
```



```
21 if (podeCadastrar){
22   listaPecas.push(nomePeca)
23   console.log('Cadastramento efetuado com sucesso')
24 } else {
25   console.log('Houve problemas para cadastrar a peça')
26 }
27
28 console.log('Listando as peças cadastradas.')
29 for (let i=0;i<listaPecas.length;i++){
30   console.log(listaPecas[i])
31 }
```

Depois de digitado o código, abram o terminal, utilizando as teclas Ctrl + ' ou acessem o menu Ver, opção Terminal como na imagem abaixo:



Com a janela do terminal aberto, digite o comando:

```
git status
```

```
git add .
```

```
git status
```

```
git commit -m 'Meu primeiro commit '
```

```
git push -u origin master
```

Isso atualizará o nosso código no github, e agora vamos aos testes. 😊



Etapa 4

Agora vamos a etapa de testes do nosso algoritmo.

Situação 1

Executem o código e respondam as questões:

Nessa situação, a peça foi cadastrada?

Quais os valores mostrados na listagem de peças?

Situação 2

Vamos inicialmente alterar o valor do peso na variável **pesoPeca**, alterem o valor de 300 para 50, depois executem o código.

Nessa situação, a peça foi cadastrada?

Quais os valores mostrados na listagem de peças?

Situação 3

Voltem o valor da variável **pesoPeca** para 300, e alterem o valor da variável **listaPecas** de [] para ['peça 1', 'peça 2', 'peça 3', 'peça 4', 'peça 5', 'peça 6', 'peça 7', 'peça 8', 'peça 9', 'peça 10', 'peça 11'], depois executem o código.

Nessa situação, a peça foi cadastrada?

Quais os valores mostrados na listagem de peças?

Situação 4

Voltem o valor da variável **listaPecas** para [], e alterem o valor da variável **nomePeca** para 'pe', depois executem o código.

Nessa situação, a peça foi cadastrada?

Quais os valores mostrados na listagem de peças?



Material extra 😊

Para quem quiser deixar seus logs do console mais bonitos, segue um link com os códigos de cores para usar no console.log.

<https://telepathy.freedesktop.org/doc/telepathy-glib/telepathy-glib-debug-ansi.html>

Para usar por exemplo a cor da fonte vermelha, usamos o seguinte código:

```
const fg_red = "\x1b[31m"  
const reset_color = "\x1b[0m"  
console.log(fg_red,"Mensagem que deseja em vermelho", reset_color)
```

A primeira parte é o código da cor vermelha, a última parte é o comando para indicar o término da cor.