

PAUCIA LIRA NUNES LISBOA



LÓGICA

Lógica de programação é a organização coesa de uma sequência de instruções voltadas à resolução de um problema, ou à criação de um software ou aplicação.

GIT

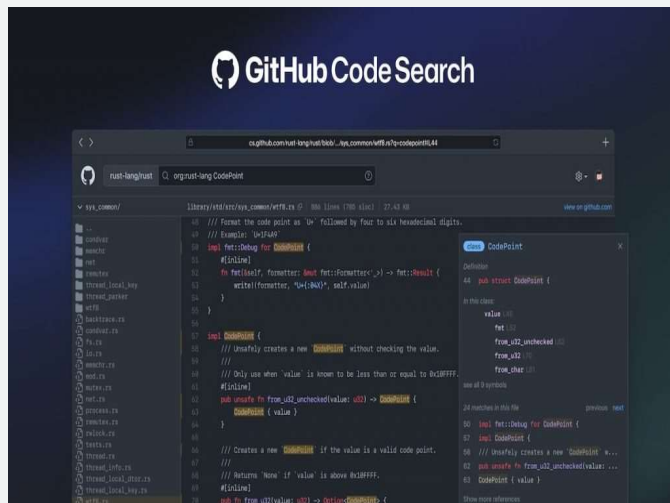
Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência.

O Git é fácil de aprender e tem uma pegada pequena com desempenho extremamente rápido.



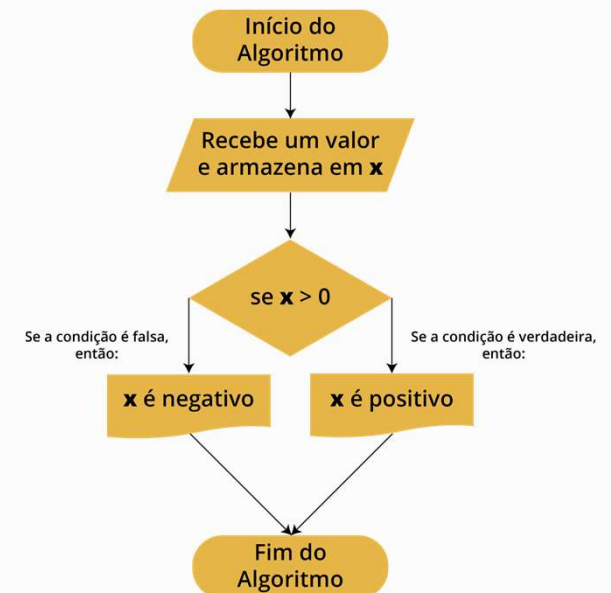
GITHUB

GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.



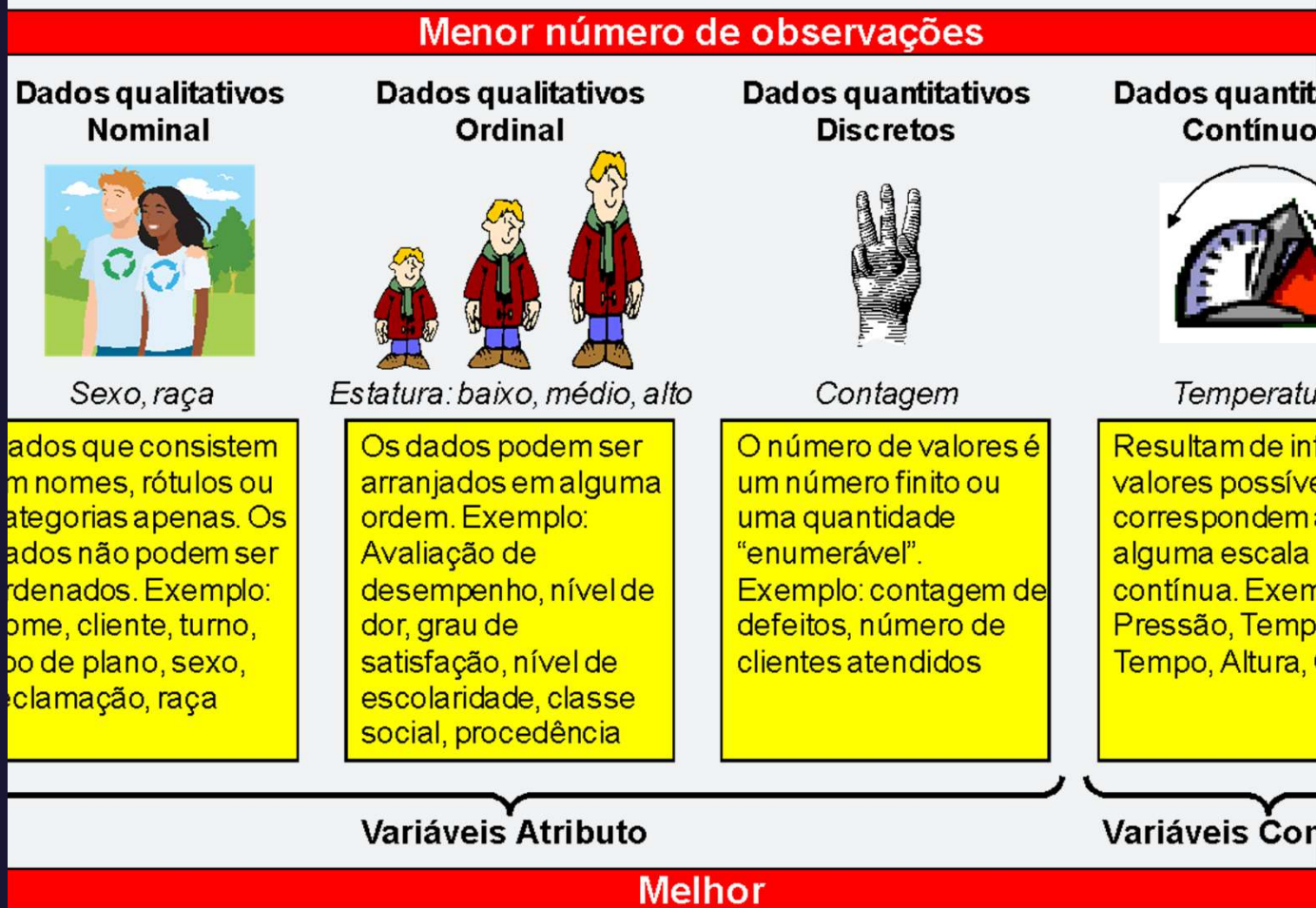
ESTRUTURA E OPERADORES

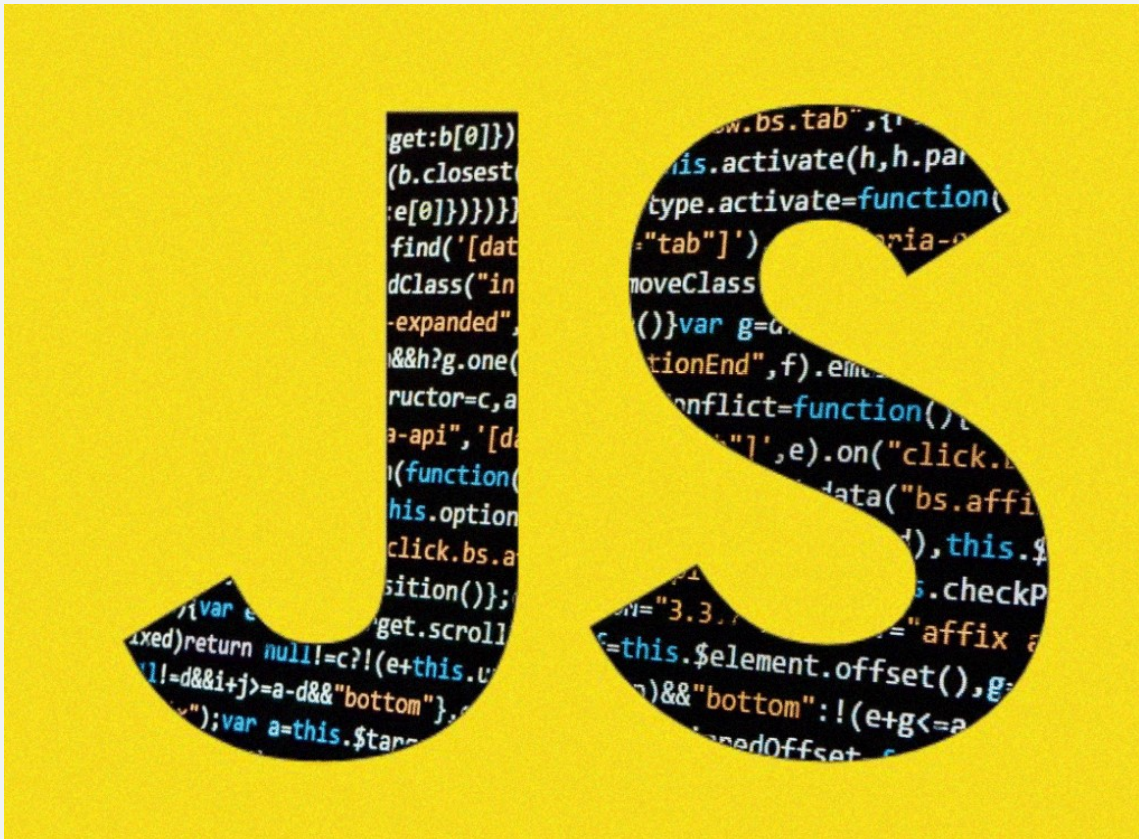
Em termos práticos, uma estrutura é um forma conveniente de agrupar vários pedaços de informação relacionados. Uma estrutura pode ser definida como um novo tipo de dados, aumentando os tipos de dados built-in da linguagem C. Isto é sempre recomendável, pois torna o código mais legível e mais simples de programar.



VARIÁVEIS

Variável é a característica de interesse que é medida em cada elemento da amostra ou população. Como o nome diz, seus valores variam de elemento para elemento. As variáveis podem ter valores numéricos ou não numéricos.





INTRODUÇÃO AO JAVASCRIPT

JavaScript é uma linguagem de script orientada a objetos, multiplataforma. É uma linguagem pequena e leve. Dentro de um ambiente de host (por exemplo, um navegador web) o JavaScript pode ser ligado aos objetos deste ambiente para prover um controle programático sobre eles.

NODE.JS

Node.js é uma tecnologia usada para executar código JavaScript fora do navegador. Com ele podemos construir aplicações web em geral, desde web sites até APIs e microsserviços. Isso é possível graças a união do ambiente de execução de JavaScript fornecido pelo próprio Node.js e o motor de interpretação e execução de JavaScript presente no Google Chrome, chamado de V8.



LÓGICA APLICADA

Lógica de programação é o modo como se escreve um programa de computador, um algoritmo. Um algoritmo é uma sequência de passos para se executar uma função. Um exemplo de algoritmo, fora da computação, é uma receita de bolo. Na receita, devem-se seguir os passos para o bolo ficar pronto e sem nenhum problema.

```
> let idade = 17;
< undefined

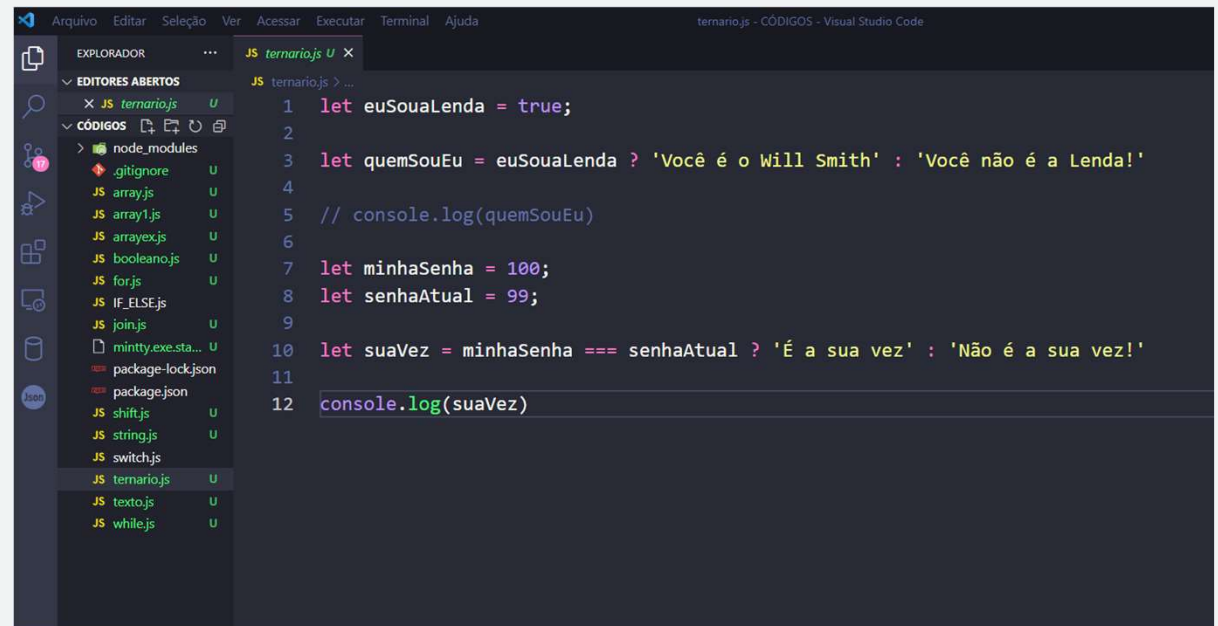
> if (idade > 15 && idade < 18 || idade > 70) {
  console.log("O voto é opcional")
}else if (idade < 16) {
  console.log("Você não pode votar");
}
O voto é opcional
< undefined

> idade = 12;
< 12

> if (idade > 15 && idade < 18 || idade > 70) {
  console.log("O voto é opcional")
}else if (idade < 16) {
  console.log("Você não pode votar");
}
Você não pode votar
< undefined
>
```

CONDICIONAL AVANÇADO

A formatação condicional facilita realçar certos valores ou tornar determinadas células fáceis de identificar. Isso altera a aparência de um intervalo de células com base em uma condição (ou critérios).



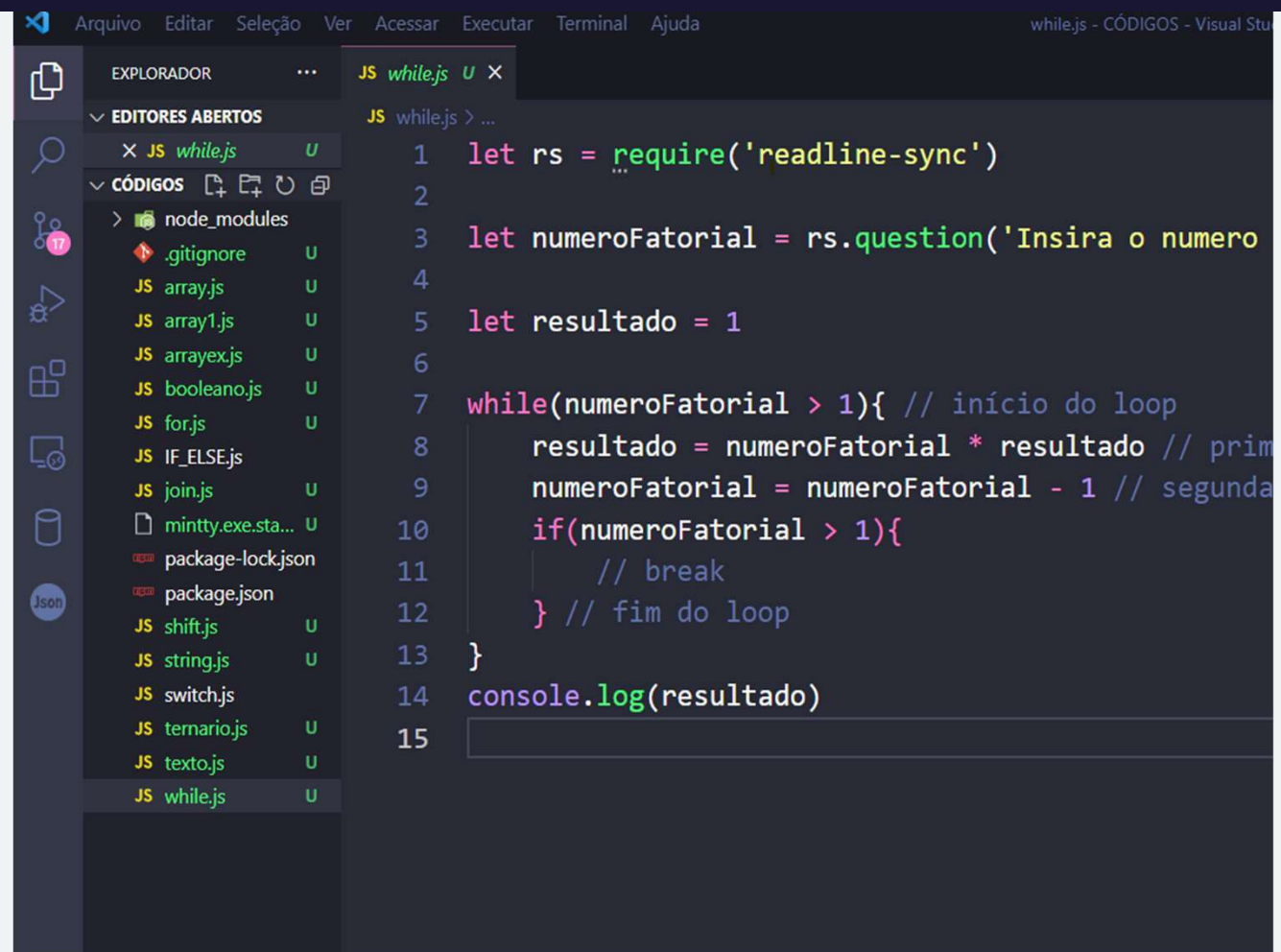
```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  ternario.js - CÓDIGOS - Visual Studio Code

EXPLORADOR
  EDITORES ABERTOS
    JS ternario.js
  CÓDIGOS
    node_modules
    .gitignore
    JS array.js
    JS array1.js
    JS arrayex.js
    JS boolean.js
    JS for.js
    JS IF_ELSE.js
    JS join.js
    minitty.exe.sta...
    package-lock.json
    package.json
    JS shift.js
    JS string.js
    JS switch.js
    JS ternario.js
    JS texto.js
    JS while.js

JS ternario.js > ...
1  let euSouaLenda = true;
2
3  let quemSouEu = euSouaLenda ? 'Você é o Will Smith' : 'Você não é a Lenda!'
4
5  // console.log(quemSouEu)
6
7  let minhaSenha = 100;
8  let senhaAtual = 99;
9
10 let suaVez = minhaSenha === senhaAtual ? 'É a sua vez' : 'Não é a sua vez!'
11
12 console.log(suaVez)
```

WHILE / LOOP

Na maioria das linguagens de programação de computadores, um loop while é uma instrução de fluxo de controle que permite que o código seja executado repetidamente com base em uma determinada condição booleana. O loop while pode ser considerado uma declaração if de repetição.

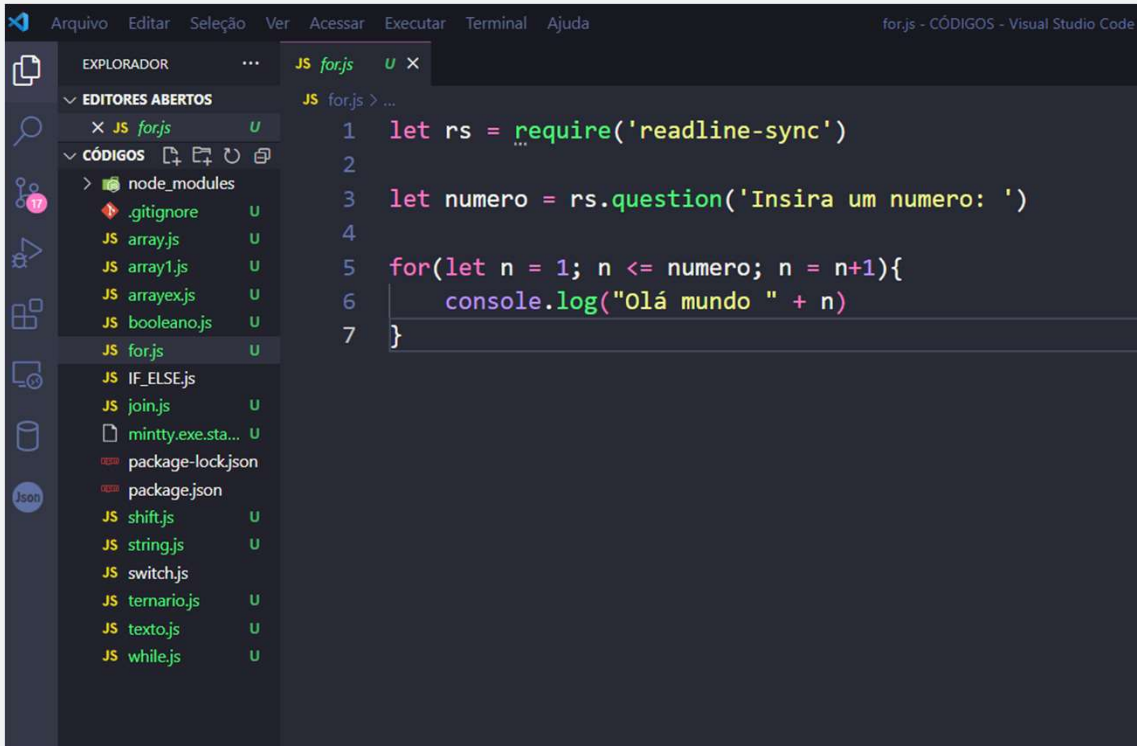


The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the file structure, including a folder named 'node_modules' and several JavaScript files. The main editor area shows the code for 'while.js'.

```
1 let rs = require('readline-sync')
2
3 let numeroFatorial = rs.question('Insira o numero
4
5 let resultado = 1
6
7 while(numeroFatorial > 1){ // início do loop
8     resultado = numeroFatorial * resultado // prim
9     numeroFatorial = numeroFatorial - 1 // segunda
10    if(numeroFatorial > 1){
11        // break
12    } // fim do loop
13 }
14 console.log(resultado)
15
```

FOR

O loop for é um comando que tem comportamento igual ao while, porém ele permite expressar contadores como parte do comando e não como lógica separada.



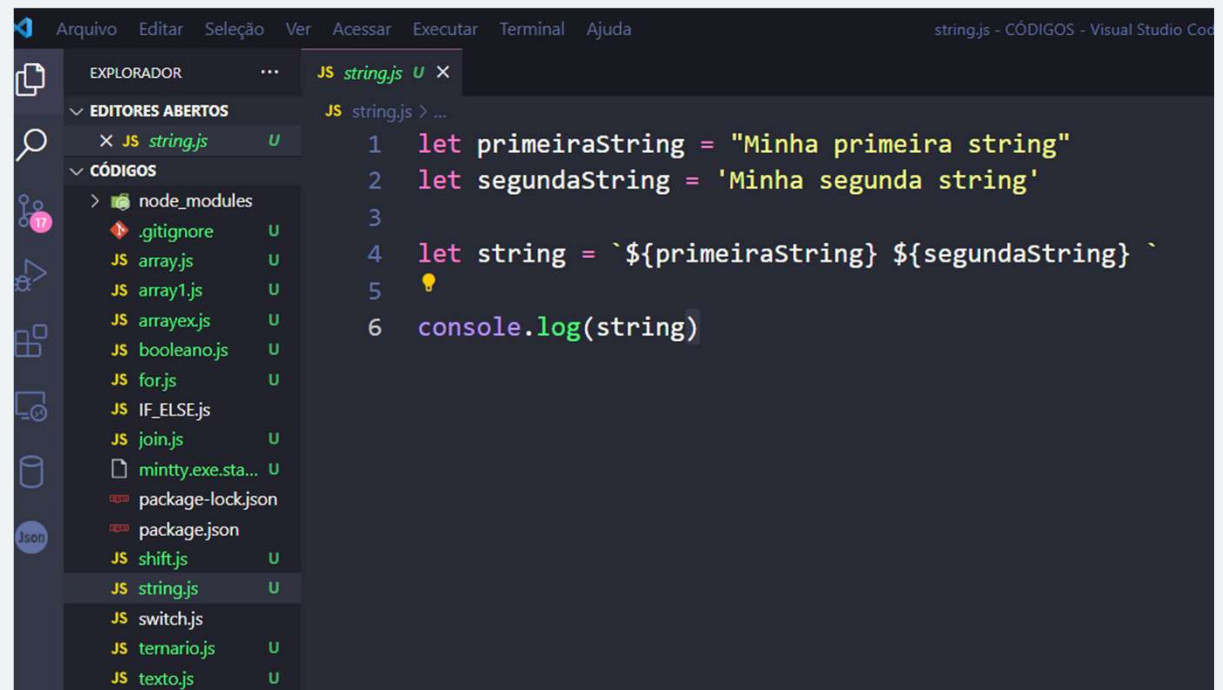
```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  for.js - CÓDIGOS - Visual Studio Code

EXPLORADOR
  EDITORES ABERTOS
    JS for.js
  CÓDIGOS
    node_modules
      .gitignore
      JS array.js
      JS array1.js
      JS arrayex.js
      JS booleano.js
      JS for.js
      JS IF_ELSE.js
      JS join.js
      minitty.exe.sta...
      package-lock.json
      package.json
      JS shift.js
      JS string.js
      JS switch.js
      JS ternario.js
      JS texto.js
      JS while.js

JS for.js
1  let rs = require('readline-sync')
2
3  let numero = rs.question('Insira um numero: ')
4
5  for(let n = 1; n <= numero; n = n+1){
6      console.log("Olá mundo " + n)
7  }
```

STRING

Uma string nada mais é que uma lista de caracteres, no qual cada elemento da lista é representado por um caractere.



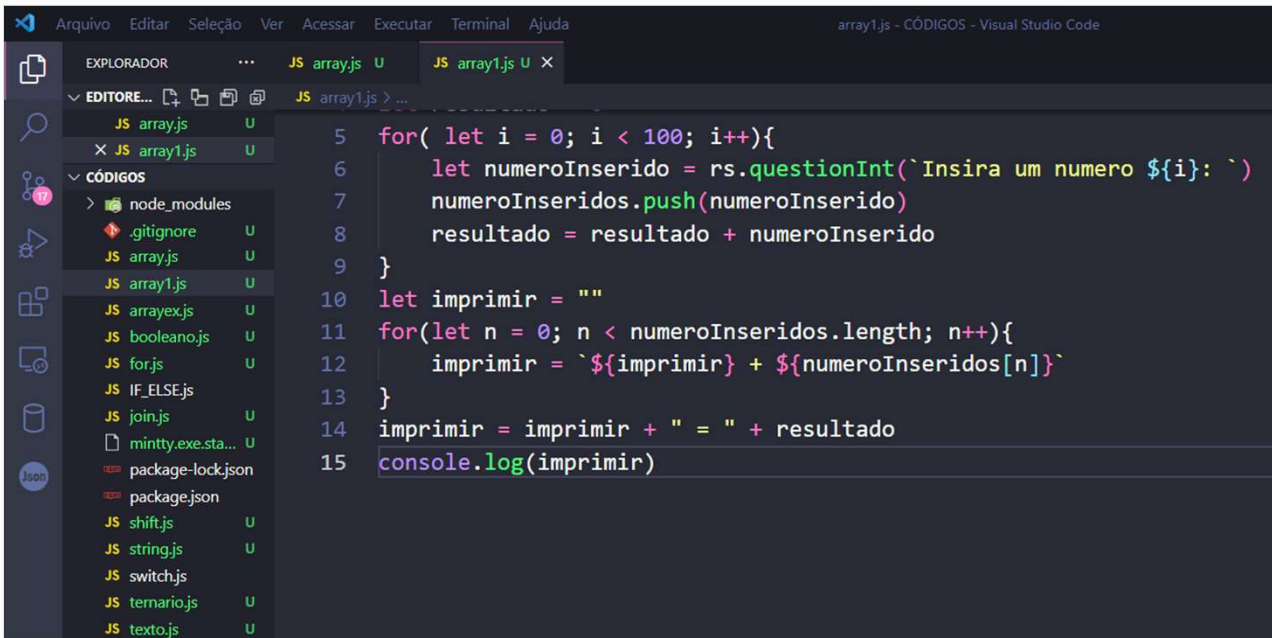
```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  string.js - CÓDIGOS - Visual Studio Cod

EXPLORADOR  JS string.js U X
EDITORES ABERTOS
  X JS string.js U
CÓDIGOS
  > node_modules
    .gitignore U
    JS array.js U
    JS array1.js U
    JS arrayex.js U
    JS booleano.js U
    JS for.js U
    JS IF_ELSE.js U
    JS join.js U
    mintty.exe.sta... U
    package-lock.json
    package.json
    JS shift.js U
    JS string.js U
    JS switch.js
    JS ternario.js U
    JS texto.js U

JS string.js > ...
1  let primeiraString = "Minha primeira string"
2  let segundaString = 'Minha segunda string'
3
4  let string = `${primeiraString} ${segundaString}`
5  ⚡
6  console.log(string)
```

ARRAY

Os arrays são a estrutura de dados mais básica da programação. Um array é uma variável que possui uma coleção de valores armazenados em sequência. Em javascript, os arrays podem guardar valores de diferentes tipos.



```
5 for( let i = 0; i < 100; i++){
6     let numeroInserido = rs.questionInt(`Insira um numero ${i}: `)
7     numeroInseridos.push(numeroInserido)
8     resultado = resultado + numeroInserido
9 }
10 let imprimir = ""
11 for(let n = 0; n < numeroInseridos.length; n++){
12     imprimir = `${imprimir} + ${numeroInseridos[n]}`
13 }
14 imprimir = imprimir + " = " + resultado
15 console.log(imprimir)
```