



SENAI

CODIFICAÇÃO PARA BACK-END



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

SUMÁRIO

- JAVASCRIPT E NODE.JS
 - INSTALAÇÃO .NET FRAMEWORK
 - INSTALAÇÃO VISUAL STUDIO CODE
 - INSTALAÇÃO NODE.JS
 - CONFIGURAÇÃO VISUAL STUDIO CODE
- JAVASCRIPT
 - INTRODUÇÃO
 - VARIÁVEIS
 - TIPOS DE DADOS



SENAI

JAVASCRIPT E NODE.JS



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

JAVASCRIPT E NODE.JS

- **JavaScript (JS):**

- Definição: JavaScript é uma linguagem de programação de alto nível, interpretada, orientada a objetos e leve. Ela é principalmente conhecida por ser a linguagem de script que é executada no navegador, permitindo a criação de páginas web interativas.

JAVASCRIPT E NODE.JS

- **Node.js:**

- Definição: Node.js é um ambiente de execução para JavaScript no lado do servidor. Ele permite que você execute o JavaScript fora do navegador, o que é essencial para construir aplicativos web do lado do servidor e realizar operações assíncronas de forma eficiente.

JAVASCRIPT E NODE.JS

- **Node.js:**

- Características: Node.js utiliza o motor V8, mecanismo de execução de JavaScript de código aberto desenvolvido pelo Google, para executar código JavaScript de forma eficiente. Possui uma arquitetura assíncrona, sendo adequado para manipulação de muitas conexões simultâneas. Vem com um grande número de módulos nativos que facilitam o desenvolvimento, e os desenvolvedores também podem criar seus próprios módulos.



SENAI

INSTALAÇÃO .NET FRAMEWORK



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

INSTALAÇÃO .NET FRAMEWORK



A screenshot of a Google search interface with a dark theme. The search bar contains the text "microsoft .net framework download". Below the search bar, there are several filter buttons: "Vídeos", "Imagens", "Shopping", "Notícias", "Livros", "Maps", "Voos", and "Finanças". The search results show approximately 42,700,000 results in 0.23 seconds. The top result is from Microsoft, with the URL "https://dotnet.microsoft.com > pt-br > download > dot...". The title of the result is "Baixar .NET Framework | Downloads oficiais gratuitos - Microsoft". The description says "Downloads para criar e executar aplicativos com o .NET Framework. Baixe o instalador da web, o instalador offline e o pacote de idiomas para o .". At the bottom of the description, there are version numbers: "4.8 · 4.7.2 · 4.5 · 4.8.1".

Google

microsoft .net framework download

Vídeos Imagens Shopping Notícias Livros Maps Voos Finanças

Aproximadamente 42.700.000 resultados (0,23 segundos)

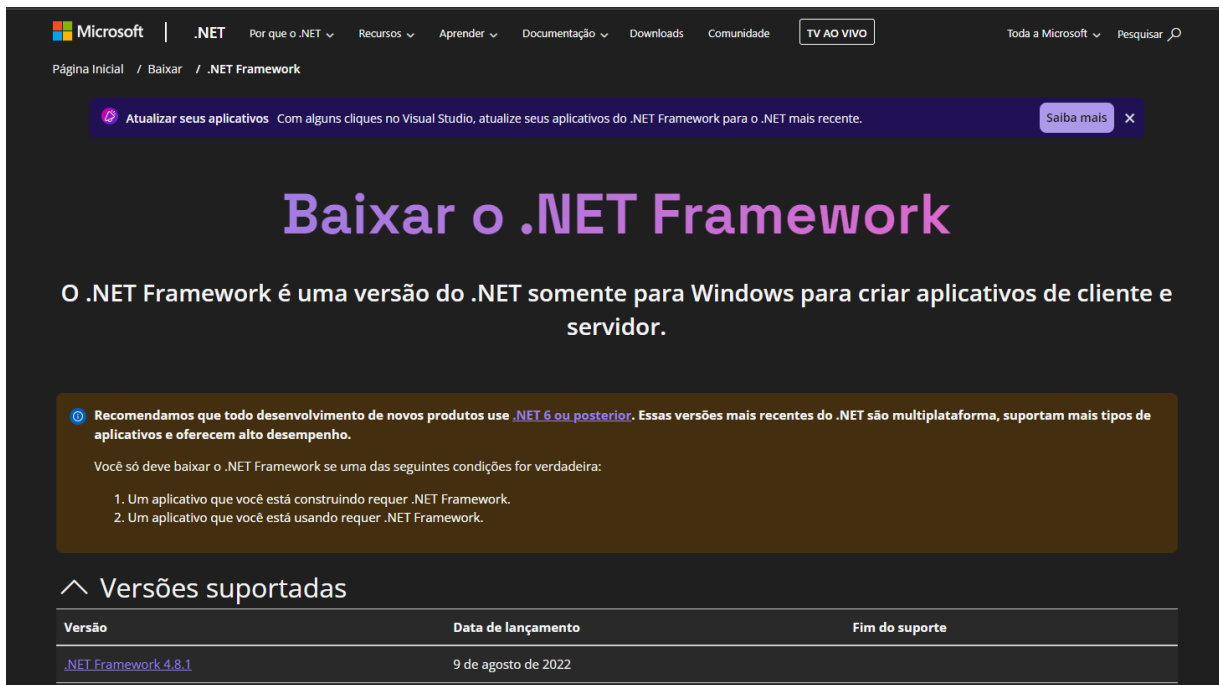
 Microsoft
https://dotnet.microsoft.com > pt-br > download > dot...

Baixar .NET Framework | Downloads oficiais gratuitos - Microsoft

Downloads para criar e executar aplicativos com o .NET Framework. Baixe o instalador da web, o instalador offline e o pacote de idiomas para o .


4.8 · 4.7.2 · 4.5 · 4.8.1

INSTALAÇÃO .NET FRAMEWORK




Microsoft | .NET Por que o .NET ▾ Recursos ▾ Aprender ▾ Documentação ▾ Downloads Comunidade TV AO VIVO Toda a Microsoft ▾ Pesquisar 🔍

Página Inicial / Baixar / .NET Framework

 **Atualizar seus aplicativos** Com alguns cliques no Visual Studio, atualize seus aplicativos do .NET Framework para o .NET mais recente. Saiba mais ✕

Baixar o .NET Framework

O .NET Framework é uma versão do .NET somente para Windows para criar aplicativos de cliente e servidor.

 Recomendamos que todo desenvolvimento de novos produtos use [.NET 6 ou posterior](#). Essas versões mais recentes do .NET são multiplataforma, suportam mais tipos de aplicativos e oferecem alto desempenho.


Você só deve baixar o .NET Framework se uma das seguintes condições for verdadeira:

1. Um aplicativo que você está construindo requer .NET Framework.
2. Um aplicativo que você está usando requer .NET Framework.

^ Versões suportadas

Versão	Data de lançamento	Fim do suporte
.NET Framework 4.8.1	9 de agosto de 2022	


INSTALAÇÃO .NET FRAMEWORK


[.NET](#)
[Por que o .NET](#)
[Recursos](#)
[Aprender](#)
[Documentação](#)
[Downloads](#)
[Comunidade](#)

TV AO VIVO


[Toda a Microsoft](#)
[Pesquisar](#)

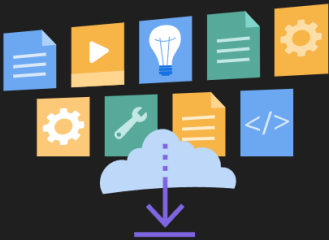
[Página Inicial](#) / [Baixar](#) / [.NET Framework](#) / **4.8.1**


Atualizar seus aplicativos Com alguns cliques no Visual Studio, atualize seus aplicativos do .NET Framework para o .NET mais recente.

Saiba mais

Baixar .NET Framework 4.8.1


 Não tem certeza do que baixar? [Consulte os downloads recomendados para a versão mais recente do .NET.](#)



Tempo de execução

Deseja executar aplicativos? O runtime inclui tudo o que você precisa para executar aplicativos/programas existentes criados com .NET Framework.

[Download .NET Framework 4.8.1 Runtime](#) →

Pacote do Desenvolvedor

Deseja criar aplicativos? O pacote de desenvolvedores é usado por desenvolvedores de software para criar aplicativos que são executados .NET Framework, normalmente usando Visual Studio.

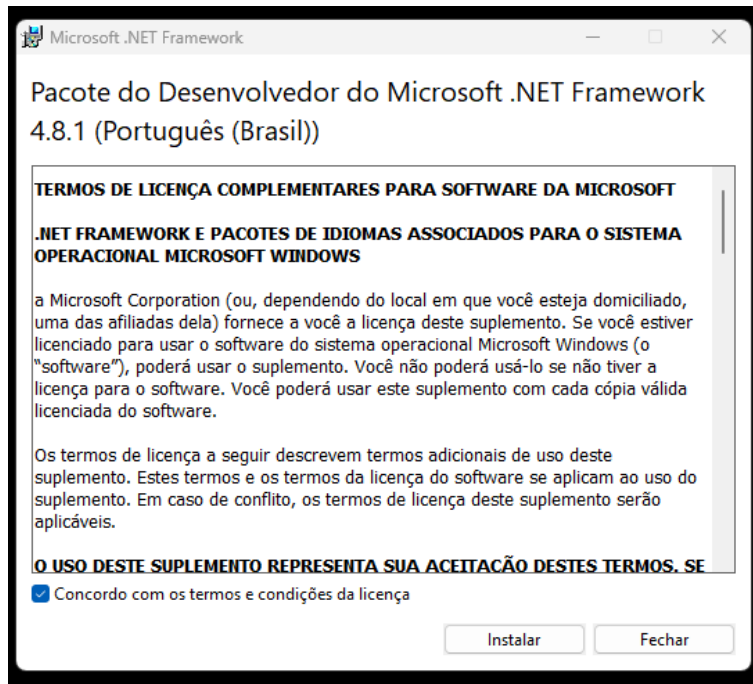
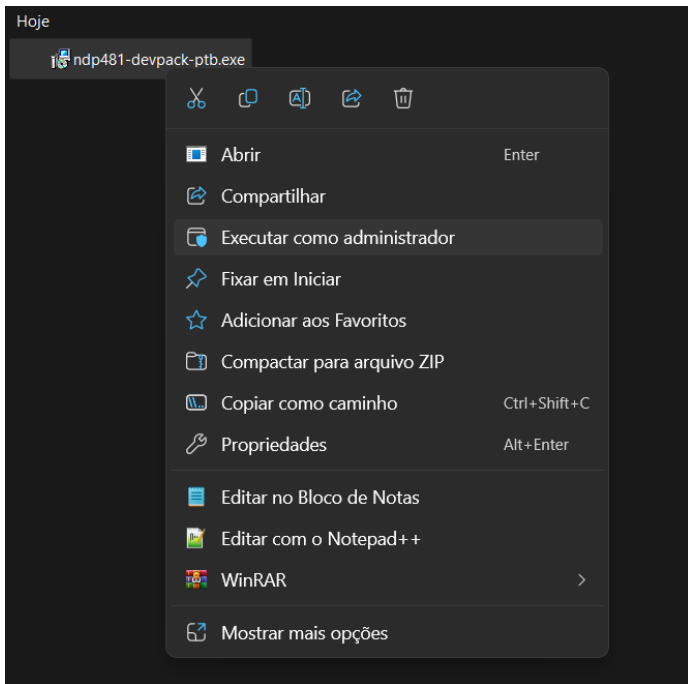
[Baixe o Pacote do Desenvolvedor .NET Framework 4.8.1](#) →

INSTALAÇÃO .NET FRAMEWORK

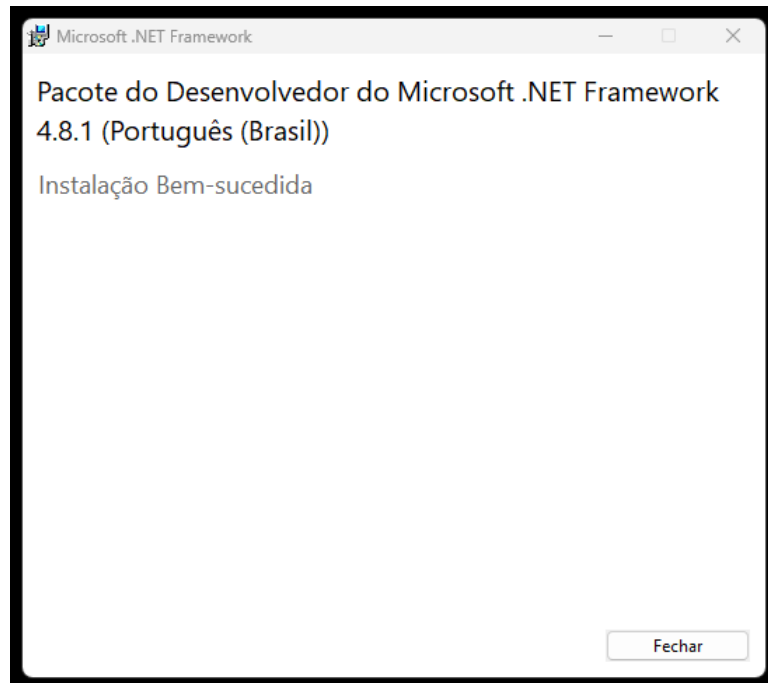
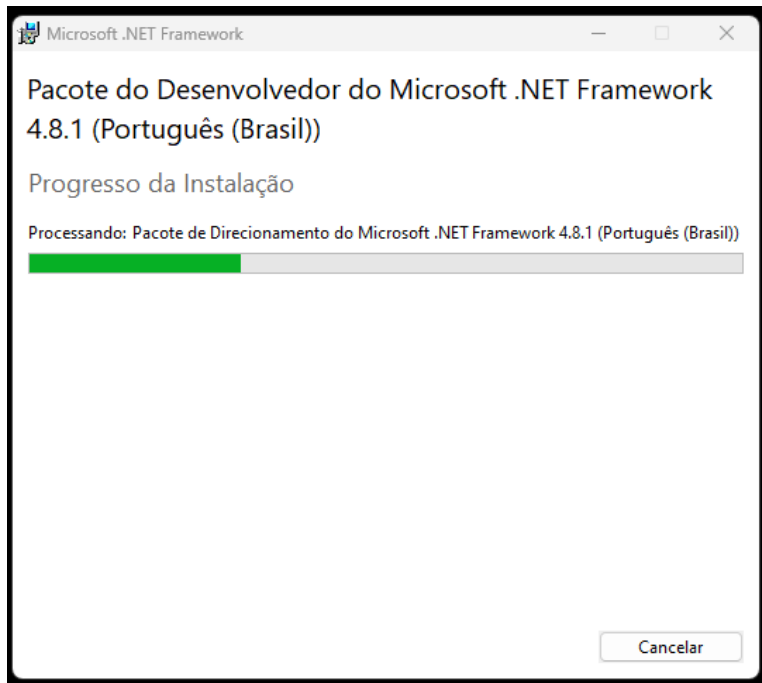
Downloads avançados

Tipo de download	Criar aplicativos - Pacote do Desenvolvedor	Executar aplicativos - Runtime
Instalador da Web	N/A	Tempo de execução
Instalador offline	Pacote do desenvolvedor	Tempo de execução
Pacotes de idiomas	<ul style="list-style-type: none"> • 中文 (简体) • 中文 (繁體) • Čeština (Česká republika) • Deutsch (Deutschland) • Español (Espanha: alfabetización internacional) • Français (France) • Italiano (Italia) • 日本語 (日本) • 한국어 (대한민국) • Polski (Polska) • Português (Brasil) • Русский (Россиа) • Türkçe (Türkiye) 	<ul style="list-style-type: none"> • العربية (المملكة العربية السعودية) • 中文 (简体) • 中文 (繁體) • Čeština (Česká republika) • Dansk (Danmark) • Deutsch (Deutschland) • Ελληνικά (Ελλάδα) • Español (Espanha: alfabetización internacional) • Suomi (Suomi) • Français (France) • עברית (ישראל) • Magyar (Magyarország) • Italiano (Italia) • 日本語 (日本) • 한국어 (대한민국) • Nederlands (Nederland) • Norsk (Norge) • Polski (Polska) • Português (Brasil) • Português (Portugal) • Русский (Россиа) • Svenska (Sverige) • Türkçe (Türkiye)

INSTALAÇÃO .NET FRAMEWORK



INSTALAÇÃO .NET FRAMEWORK





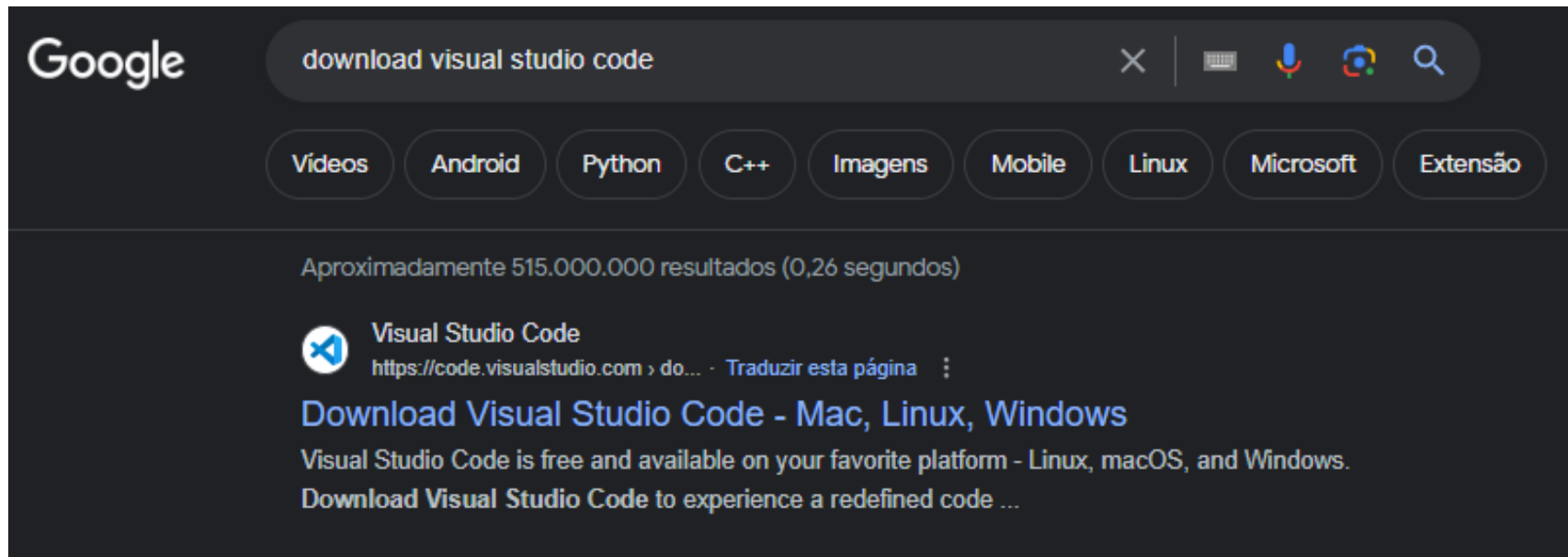
SENAI

INSTALAÇÃO VISUAL STUDIO CODE

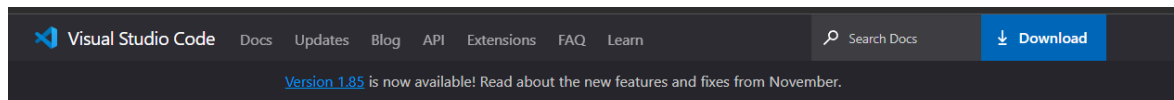


Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

INSTALAÇÃO VISUAL STUDIO CODE

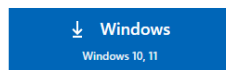


INSTALAÇÃO VISUAL STUDIO CODE

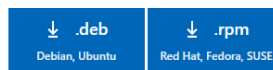


Download Visual Studio Code

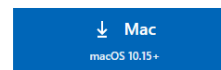
Free and built on open source. Integrated Git, debugging and extensions.



User Installer [x64](#) [Arm64](#)
System Installer [x64](#) [Arm64](#)
.zip [x64](#) [Arm64](#)
CLI [x64](#) [Arm64](#)

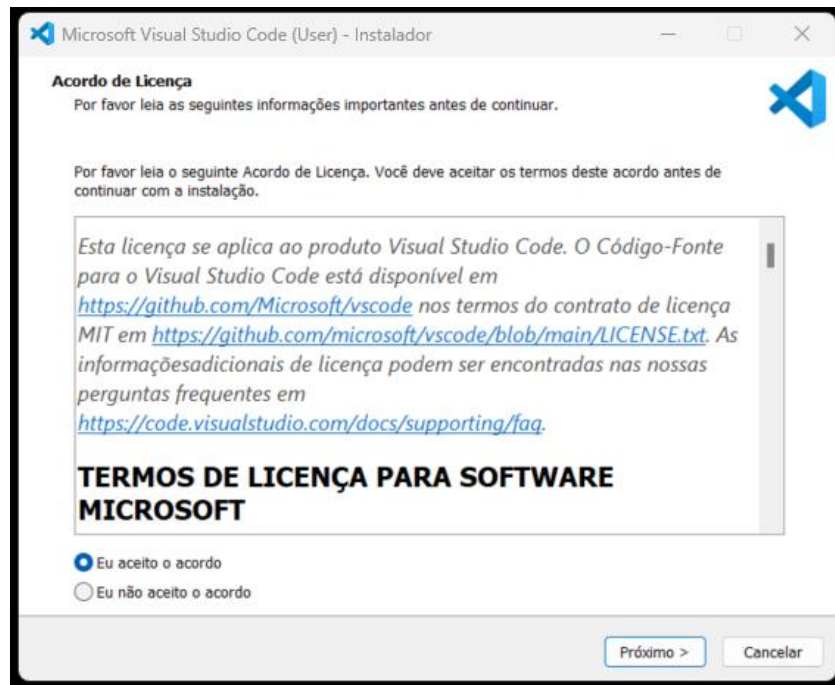
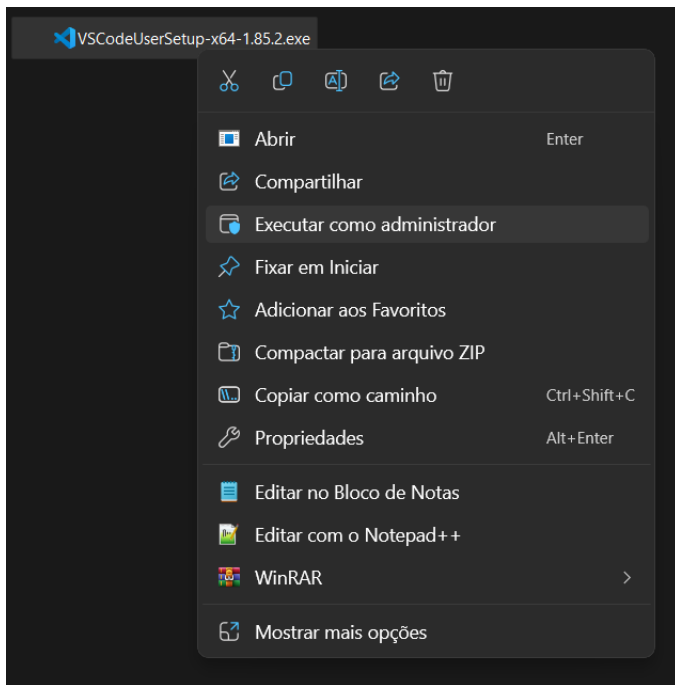


.deb [x64](#) [Arm32](#) [Arm64](#)
.rpm [x64](#) [Arm32](#) [Arm64](#)
.tar.gz [x64](#) [Arm32](#) [Arm64](#)
Snap [Snap Store](#)
CLI [x64](#) [Arm32](#) [Arm64](#)

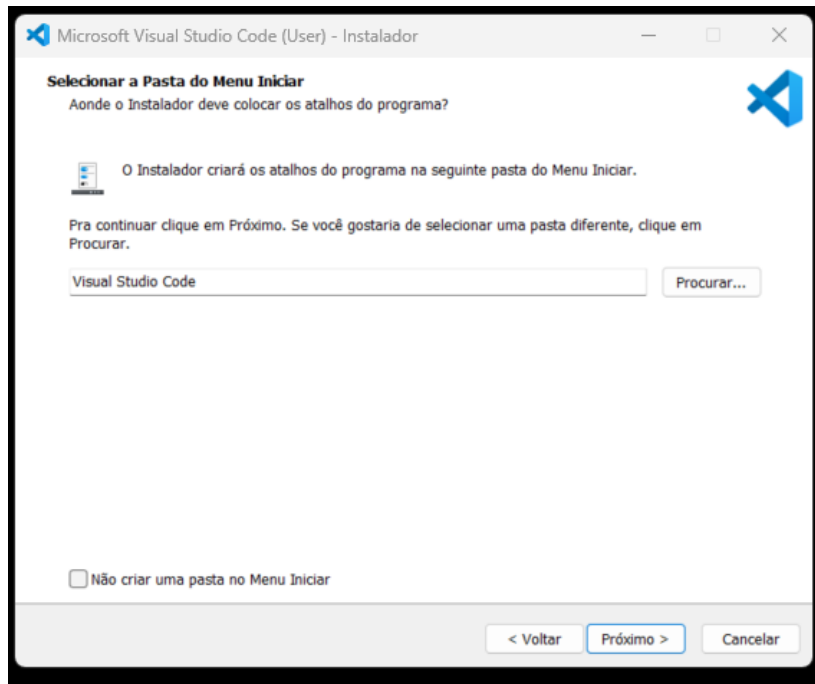
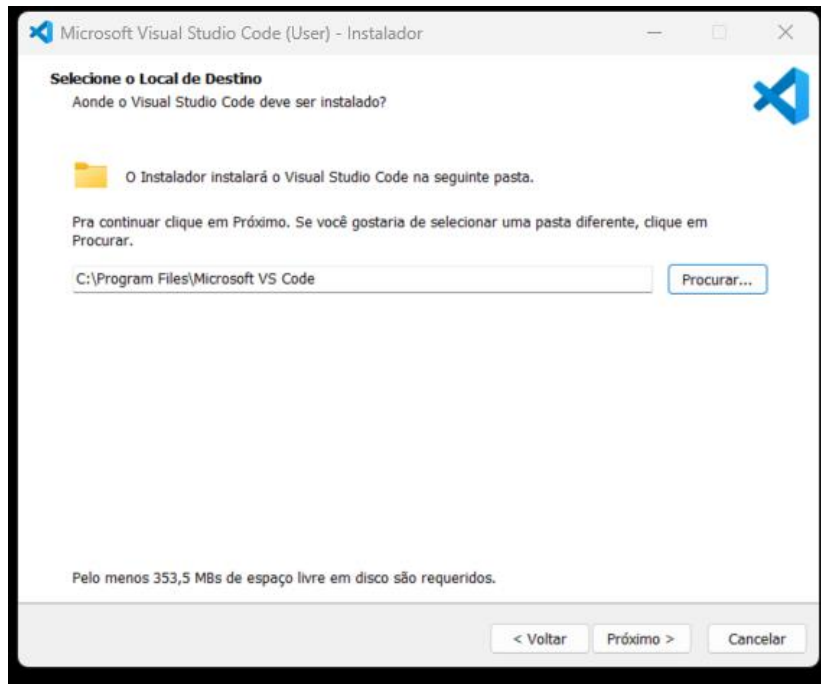


.zip [Intel chip](#) [Apple silicon](#) [Universal](#)
CLI [Intel chip](#) [Apple silicon](#)

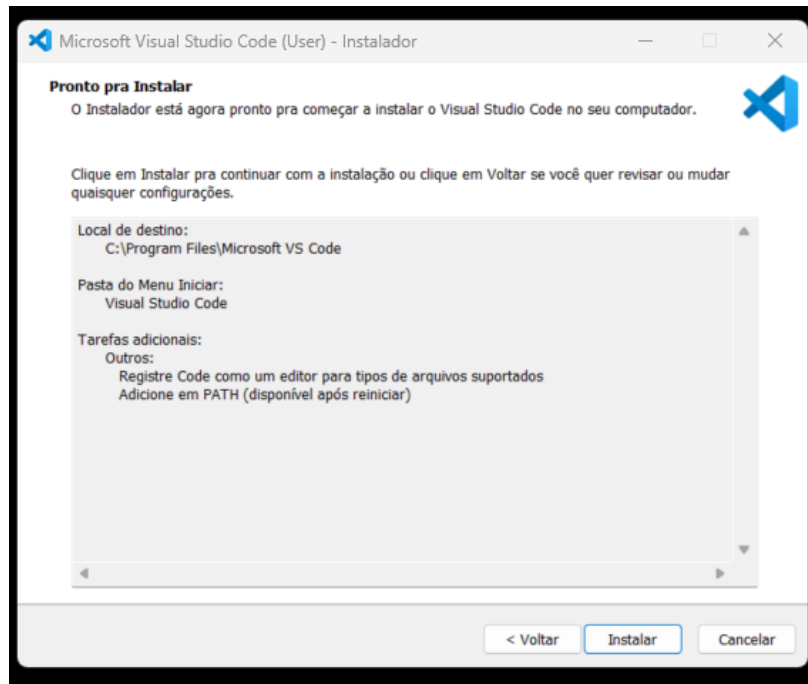
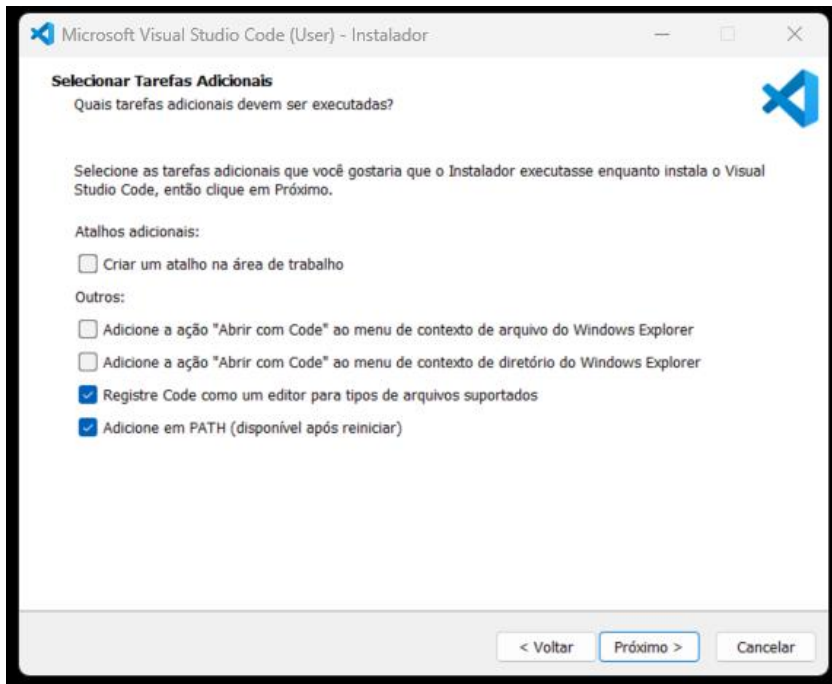
INSTALAÇÃO VISUAL STUDIO CODE



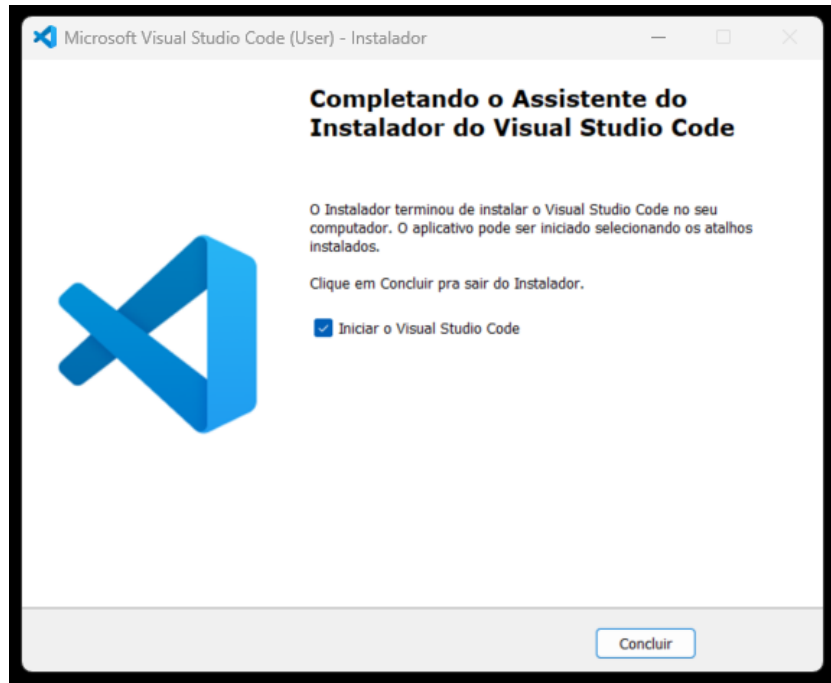
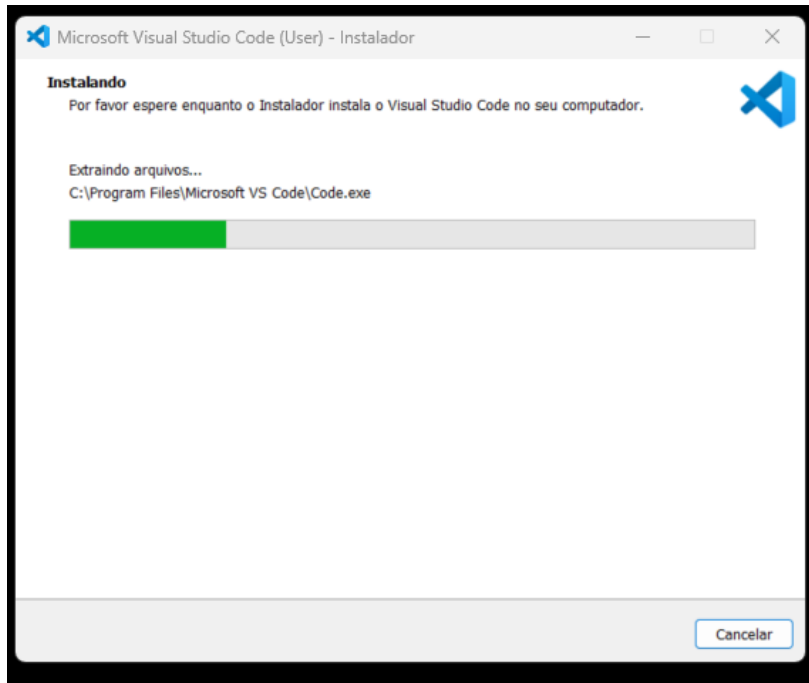
INSTALAÇÃO VISUAL STUDIO CODE



INSTALAÇÃO VISUAL STUDIO CODE



INSTALAÇÃO VISUAL STUDIO CODE





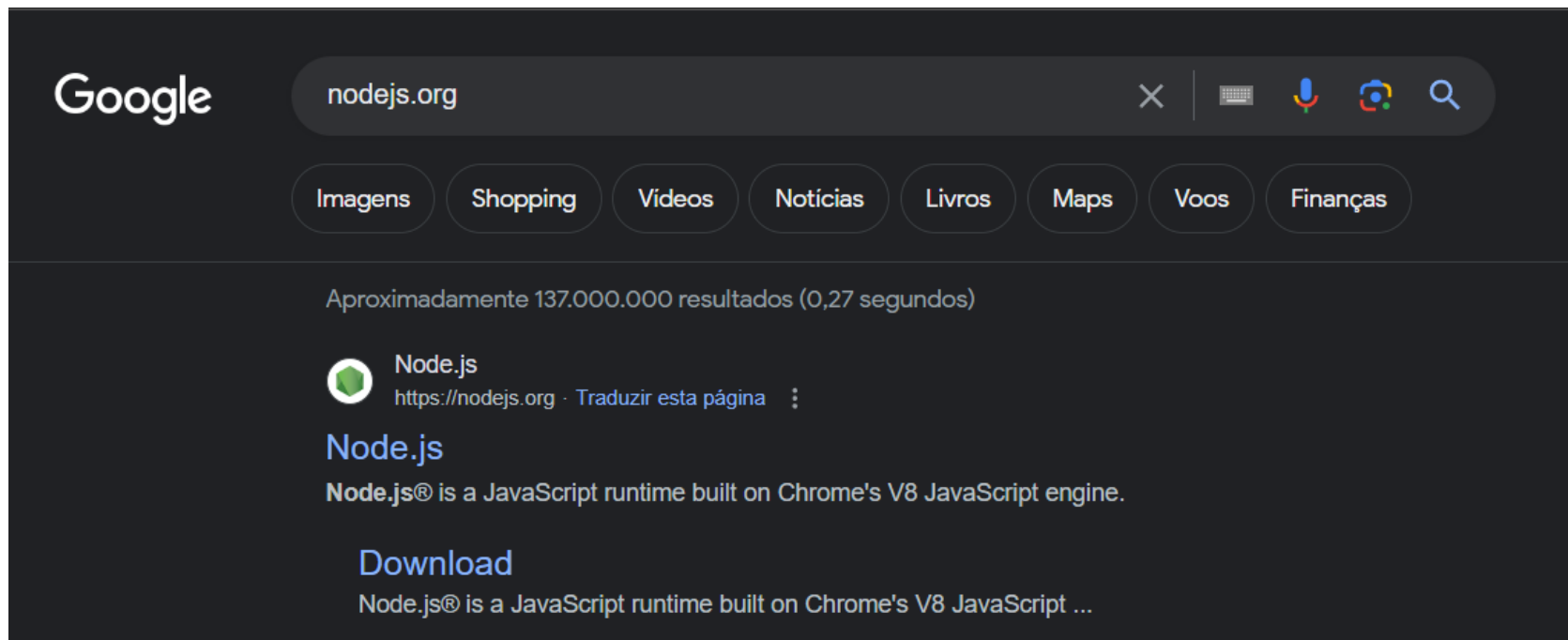
SENAI

INSTALAÇÃO NODE.JS



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

INSTALAÇÃO NODE.JS




A screenshot of a Google search interface with a dark theme. The search bar at the top contains the text "nodejs.org". Below the search bar, there are several category buttons: "Imagens", "Shopping", "Videos", "Noticias", "Livros", "Maps", "Voos", and "Finanças". The search results show approximately 137,000,000 results in 0.27 seconds. The first result is for "Node.js" with the URL "https://nodejs.org" and a link to "Traduzir esta página". Below the result, the text "Node.js" is displayed in a larger font, followed by a description: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." At the bottom, there is a "Download" link and a truncated description: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript ...".

Google

nodejs.org

Imagens Shopping Videos Noticias Livros Maps Voos Finanças

Aproximadamente 137.000.000 resultados (0,27 segundos)

 Node.js
<https://nodejs.org> · Traduzir esta página

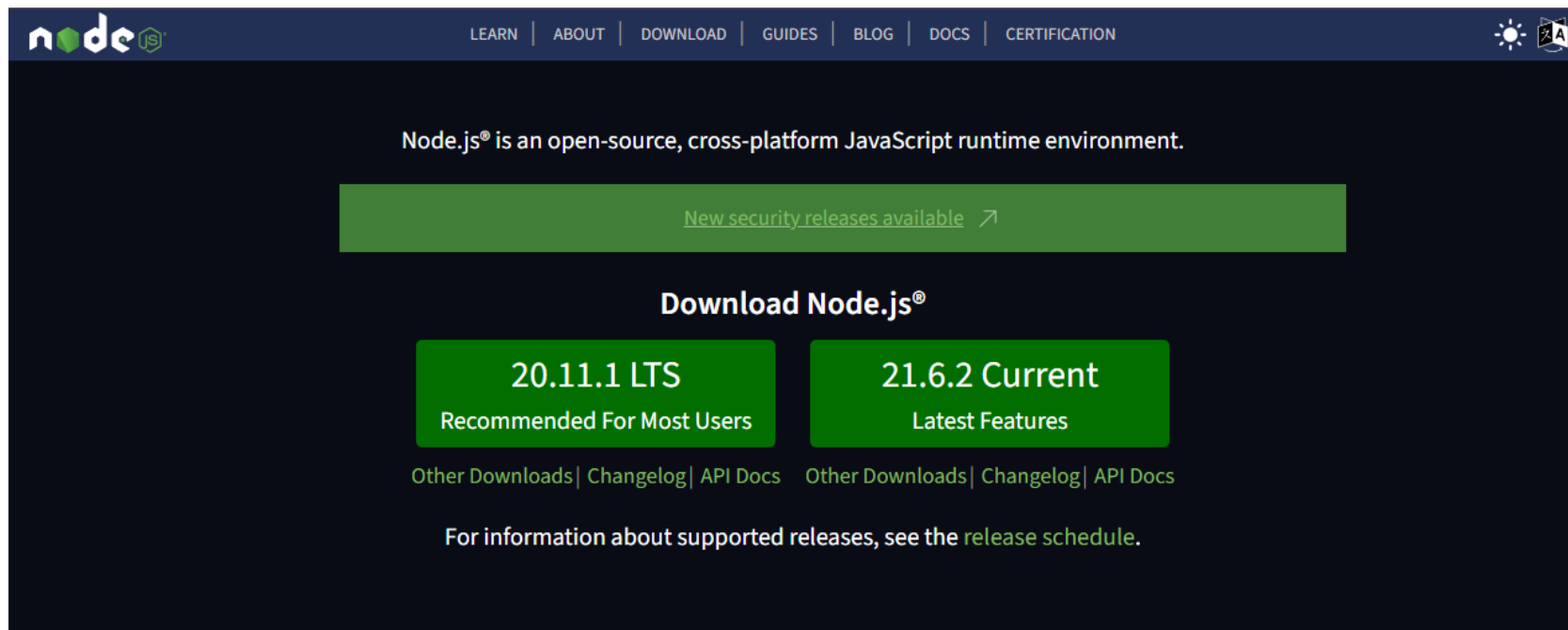
Node.js

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript ...

INSTALAÇÃO NODE.JS



The screenshot shows the Node.js website with a dark blue header. The header contains the Node.js logo on the left, a navigation menu with links: LEARN | ABOUT | DOWNLOAD | GUIDES | BLOG | DOCS | CERTIFICATION in the center, and a sun icon and a shield icon on the right. The main content area is dark blue. It starts with the text "Node.js® is an open-source, cross-platform JavaScript runtime environment." followed by a green banner with the text "New security releases available" and an arrow icon. Below this is the heading "Download Node.js®". There are two green buttons: "20.11.1 LTS" with the subtext "Recommended For Most Users" and "21.6.2 Current" with the subtext "Latest Features". Below the buttons are two links: "Other Downloads | Changelog | API Docs" repeated twice. At the bottom, it says "For information about supported releases, see the release schedule."

nodejs

LEARN | ABOUT | DOWNLOAD | GUIDES | BLOG | DOCS | CERTIFICATION

Node.js® is an open-source, cross-platform JavaScript runtime environment.

[New security releases available](#) ↗

Download Node.js®

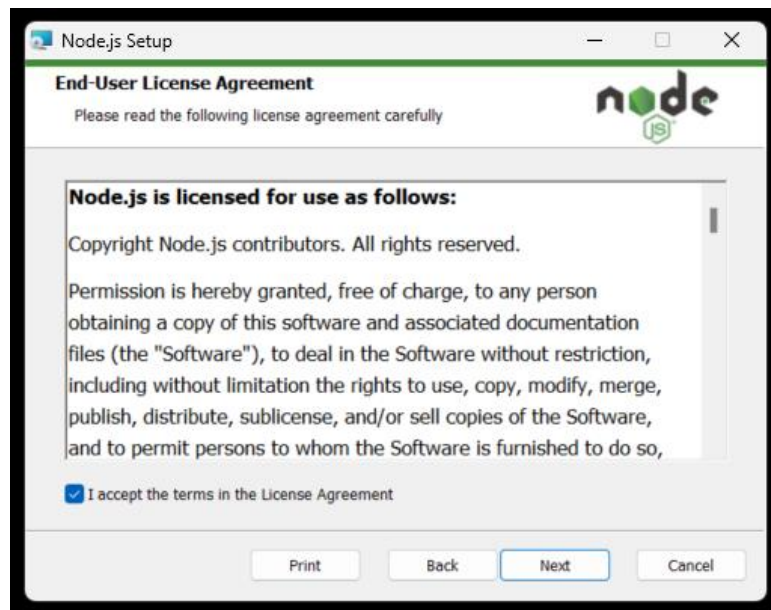
20.11.1 LTS
Recommended For Most Users

21.6.2 Current
Latest Features

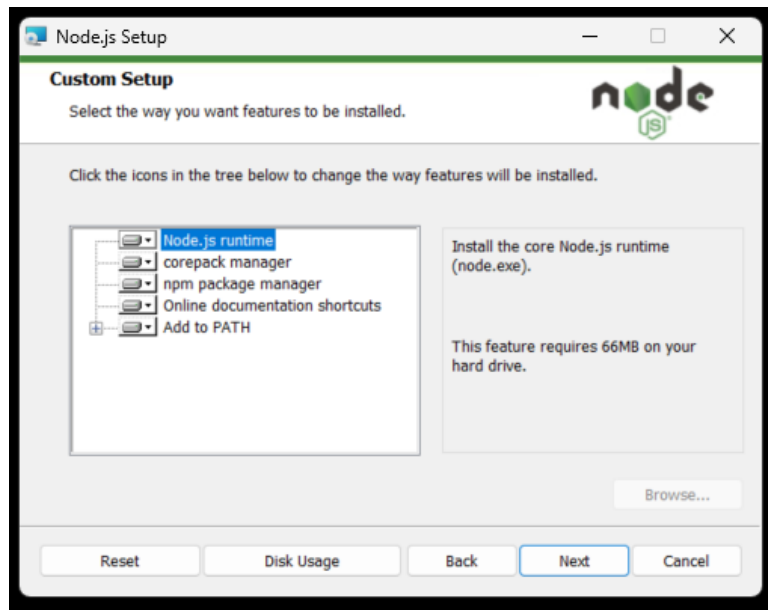
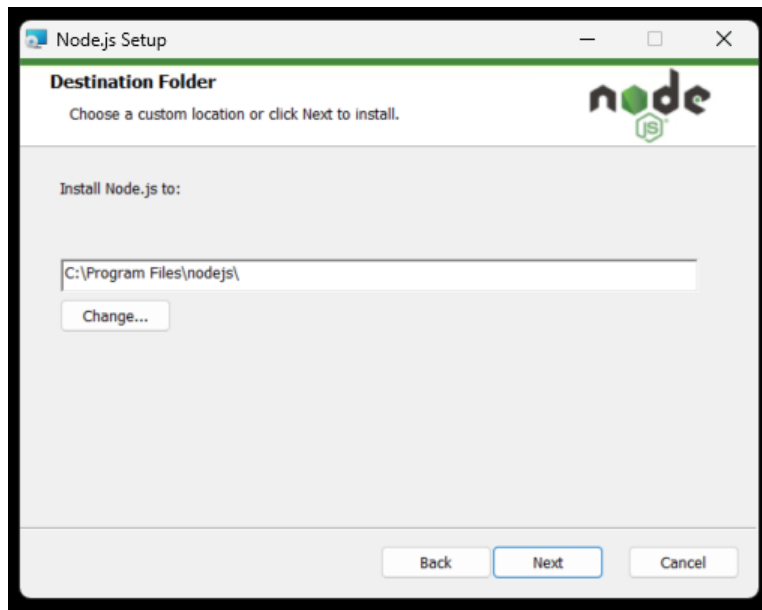
[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

For information about supported releases, see the [release schedule](#).

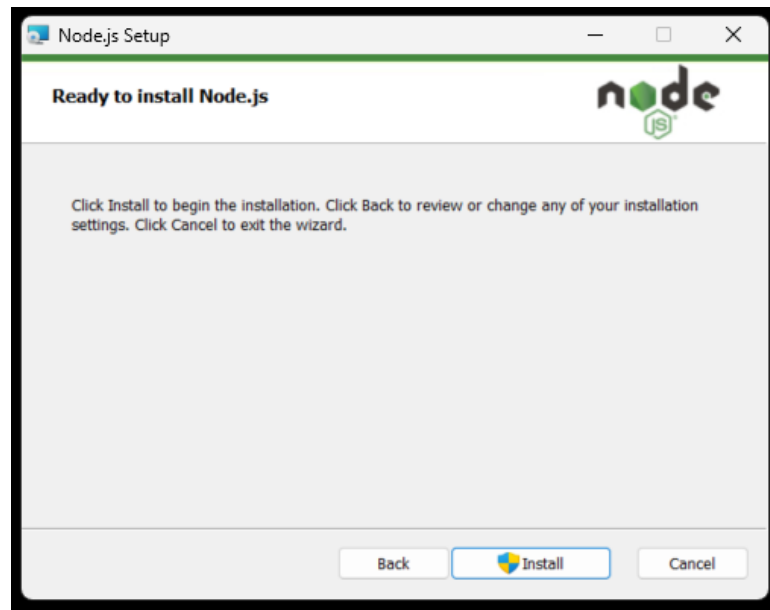
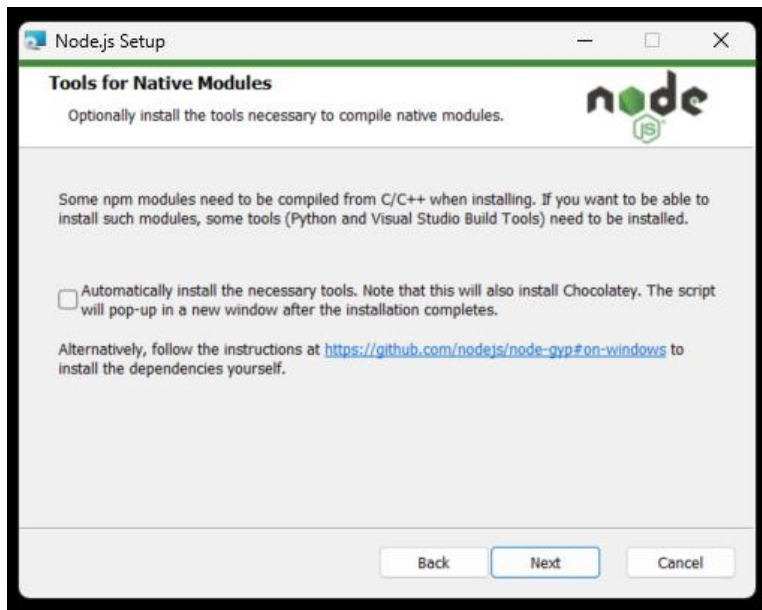
INSTALAÇÃO NODE.JS



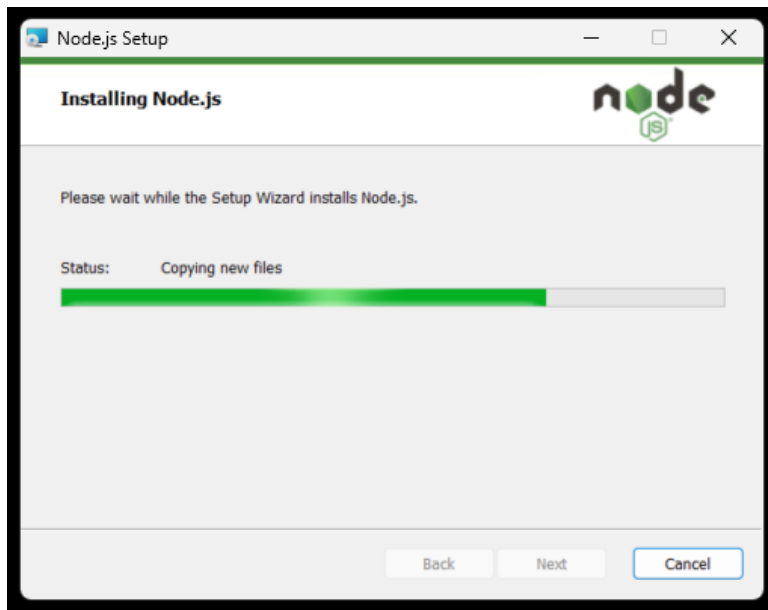
INSTALAÇÃO NODE.JS



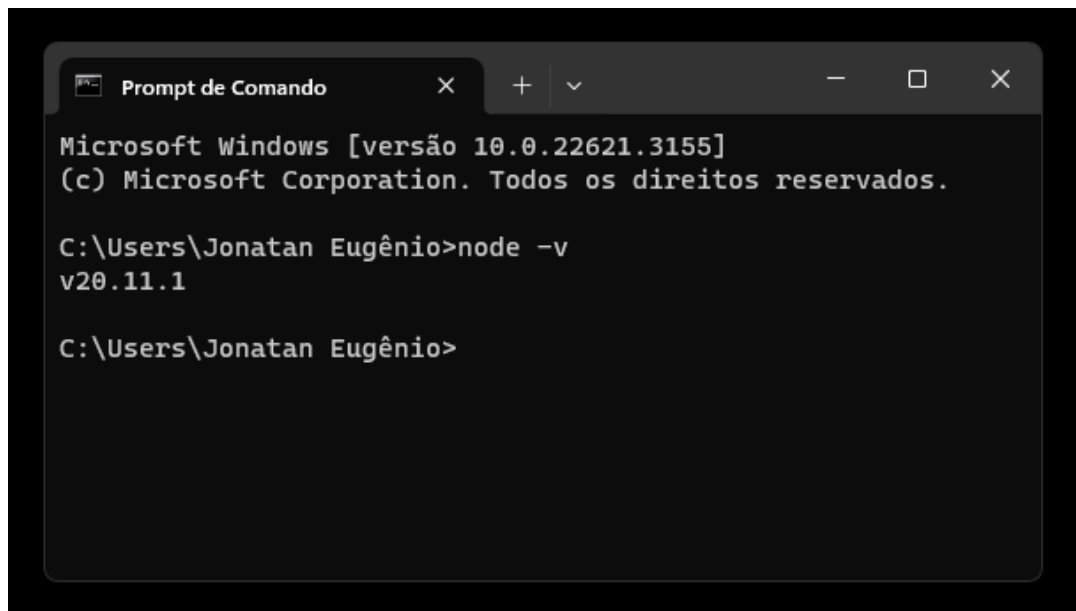
INSTALAÇÃO NODE.JS



INSTALAÇÃO NODE.JS



INSTALAÇÃO NODE.JS



```
Prompt de Comando
Microsoft Windows [versão 10.0.22621.3155]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Jonatan Eugênio>node -v
v20.11.1

C:\Users\Jonatan Eugênio>
```



SENAI

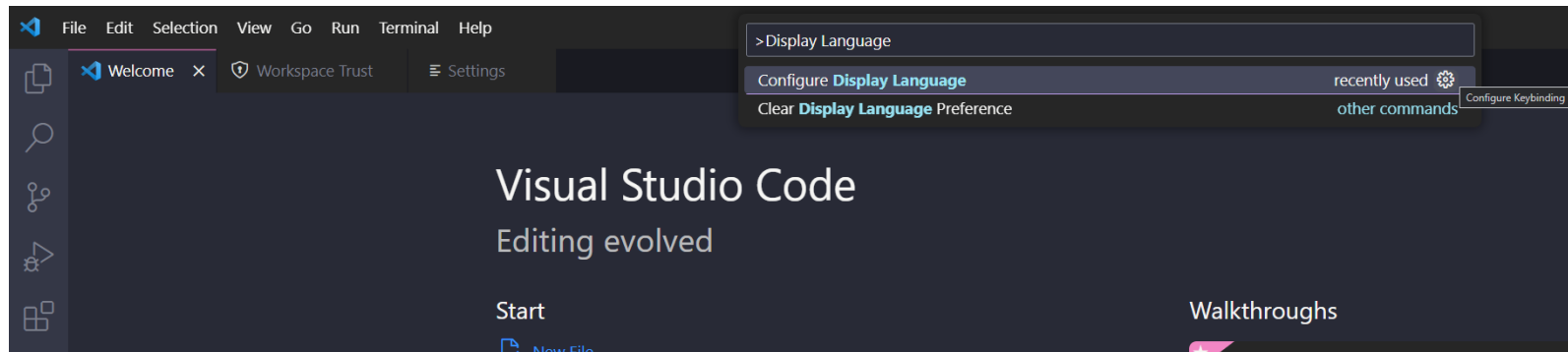
CONFIGURAÇÃO VISUAL STUDIO CODE



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

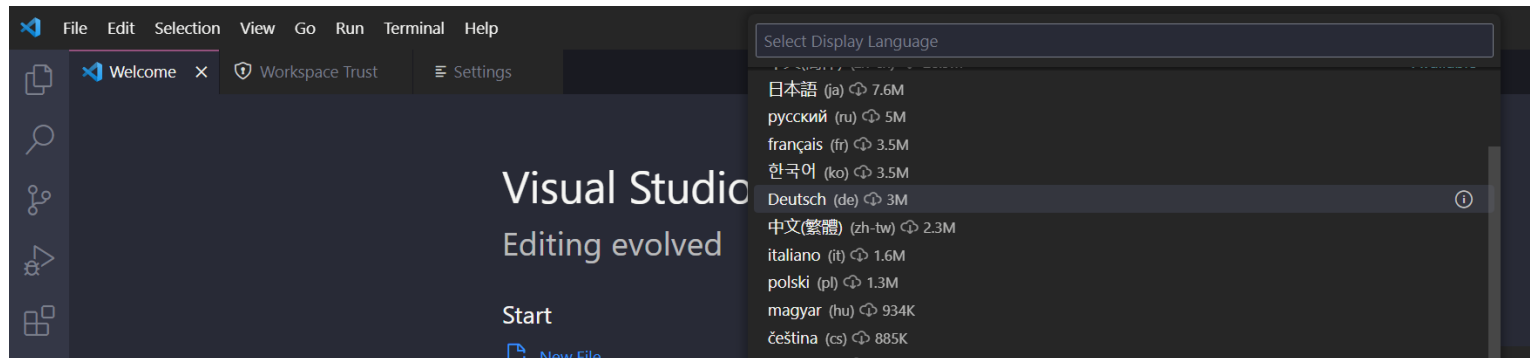
CONFIGURAÇÃO VISUAL STUDIO CODE

- Se a versão estiver em Inglês. Pressione **CTRL + SHIFT + P** e digite **Display Language**.



CONFIGURAÇÃO VISUAL STUDIO CODE

- Escolha a opção **Português (Brasil)** e reinicie o Visual Studio Code.





SENAI

JAVASCRIPT



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

JAVASCRIPT

■ Variáveis:

- Uma variável é um espaço de armazenamento com um nome simbólico (um identificador) que é associado a um valor ou a um conjunto de valores. As variáveis são usadas para armazenar e manipular dados durante a execução de um programa, elas são declaradas usando as palavras-chaves **var**, **let**, ou **const**, seguida pelo nome da variável. Cada variável tem um tipo de dado que determina o tipo de valores que ela pode armazenar, como inteiros, decimais, caracteres, booleanos, etc.

JAVASCRIPT

- **Nomenclatura de Variáveis:**
 - Devem começar com uma letra, \$, ou _.
 - Podem conter letras, números, \$ ou _.
 - Sensíveis a maiúsculas e minúsculas.

JAVASCRIPT

- **Convenções de Nomenclatura:**

- Boas práticas recomendam o uso de camelCase para nomes de variáveis, começando com uma letra minúscula e usando maiúsculas para indicar palavras subsequentes. **Exemplo: nomeDaVariavel.**
- Escolha nomes descritivos que indiquem o propósito da variável.


JAVASCRIPT

- **Var:**

- A palavra-chave **var** é a maneira tradicional de declarar variáveis, mas possui escopo de função, o que pode levar a comportamentos inesperados em alguns casos, isso se dá porque essas variáveis são visíveis em toda a função em que são declaradas, independentemente de blocos internos.

JAVASCRIPT

■ Var:



```
1 // Exemplo de declaração de var
2 function exemploVar() {
3     if (true) {
4         var valorNumerico = 10;
5         console.log(valorNumerico); // Exibe o valor 10 no console
6     }
7     console.log(valorNumerico); // Exibe o valor 10 no console
8 }
9 exemploVar();
```

JAVASCRIPT

■ Var:

```
PROBLEMAS  SAÍDA  TERMINAL  PORTAS

● PS C:\DevFullStack\Javascript\02_VariaveisJS> node scripts.js
10
10
○ PS C:\DevFullStack\Javascript\02_VariaveisJS> 
```


JAVASCRIPT

- **Let:**

- A palavra-chave **let** é a maneira moderna e introduz o conceito de escopo de bloco, o que ajuda a evitar problemas comuns associados ao escopo de função, isso se dá porque essas variáveis são visíveis apenas dentro do bloco em que são declaradas.

JAVASCRIPT

■ Let:



```
1 // Exemplo de declaração de let
2 function exemploLet() {
3     if (true) {
4         let valorNumerico = 10;
5         console.log(valorNumerico); // Exibe o valor 10 no console
6     }
7     console.log(valorNumerico); // Erro: ReferenceError: valorNumerico is not defined
8 }
9 exemploLet();
10
```


JAVASCRIPT

■ Let:

```
PROBLEMAS  SAÍDA  TERMINAL  PORTAS

⊗ PS C:\DevFullStack\Javascript\02_VariaveisJS> node scripts.js
10
C:\DevFullStack\Javascript\02_VariaveisJS\scripts.js:7
  console.log(valorNumerico); // Exibe o valor 10 no console
          ^
ReferenceError: valorNumerico is not defined
    at exemploLet (C:\DevFullStack\Javascript\02_VariaveisJS\scripts.js:7:17)
    at Object.<anonymous> (C:\DevFullStack\Javascript\02_VariaveisJS\scripts.js:9:1)
    at Module._compile (node:internal/modules/cjs/loader:1376:14)
    at Module._extensions..js (node:internal/modules/cjs/loader:1435:10)
    at Module.load (node:internal/modules/cjs/loader:1207:32)
    at Module._load (node:internal/modules/cjs/loader:1023:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:135:12)
    at node:internal/main/run_main_module:28:49

Node.js v20.11.1
○ PS C:\DevFullStack\Javascript\02_VariaveisJS> 
```

JAVASCRIPT

- **Const:**

- A palavra-chave **const** é utilizada para a declaração de uma variável constante cujo valor não pode ser reatribuído. Dessa forma, o valor de uma constante não pode ser alterado após a atribuição inicial.

JAVASCRIPT

■ Const:

```
1 // Exemplo de declaração de const
2 function exemploConst() {
3     if (true) {
4         const valorNumerico = 10;
5         console.log(valorNumerico); // Exibe o valor 10 no console
6     }
7     console.log(valorNumerico); // Erro: ReferenceError: valorNumerico is not defined
8 }
9 exemploConst();
10
```

JAVASCRIPT

■ Const:

```
PROBLEMAS  SAÍDA  TERMINAL  PORTAS

PS C:\DevFullStack\Javascript\02_VariaveisJS> node scripts.js
10
C:\DevFullStack\Javascript\02_VariaveisJS\scripts.js:7
  console.log(valorNumerico); // Erro: ReferenceError: valorNumerico is not defined
        ^

ReferenceError: valorNumerico is not defined
    at exemploConst (C:\DevFullStack\Javascript\02_VariaveisJS\scripts.js:7:17)
    at Object.<anonymous> (C:\DevFullStack\Javascript\02_VariaveisJS\scripts.js:9:1)
    at Module._compile (node:internal/modules/cjs/loader:1376:14)
    at Module._extensions..js (node:internal/modules/cjs/loader:1435:10)
    at Module.load (node:internal/modules/cjs/loader:1207:32)
    at Module._load (node:internal/modules/cjs/loader:1023:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:135:12)
    at node:internal/main/run_main_module:28:49

Node.js v20.11.1
PS C:\DevFullStack\Javascript\02_VariaveisJS> |
```

JAVASCRIPT

- **Tipos de Dados:**

- Cada variável possui um tipo de dado, que determina o tipo de valor que ela pode armazenar. JavaScript é uma linguagem de programação dinamicamente tipada, o que significa que você não precisa declarar explicitamente o tipo de uma variável ao criá-la.

JAVASCRIPT

- **Tipos de Dados:**

- **Number:** Representa números inteiros ou de ponto flutuante.



```
1 // Tipo de Dados - Number (Número): Representa números inteiros ou de ponto flutuante.  
2 let inteiro = 5;  
3 let decimal = 12.7;  
4 let inteiroNegativo = -123;
```

JAVASCRIPT

- **Tipos de Dados:**

- **String:** Representa sequências de caracteres.



```
1 // Tipo de Dados - String (Cadeia de caracteres): Representa sequências de caracteres.  
2 let texto1 = "texto1";  
3 let texto2 = 'texto2';  
4 let texto3 = `texto3`;
```

JAVASCRIPT

- **Tipos de Dados:**

- **Boolean:** Representa valores lógicos true ou false.



```
1 // Tipo de Dados - Boolean (Booleano): Representa valores lógicos true ou false.  
2 let verdadeiro = true;  
3 let falso = false;
```


JAVASCRIPT

■ Tipos de Dados:

- **Undefined:** Representa uma variável que foi declarada, mas ainda não recebeu um valor.



```
1 // Tipo de Dados - Undefined (Indefinido): Representa uma variável que foi declarada, mas ainda não recebeu um valor.  
2 let indefinido;  
3 let indefinido2 = undefined;
```

JAVASCRIPT

- **Tipos de Dados:**

- **Null:** Representa a ausência intencional de qualquer valor ou objeto.



```
1 // Tipo de Dados - Null (Nulo): Representa a ausência intencional de qualquer valor ou objeto.  
2 let nulo = null;
```

JAVASCRIPT

- **Tipos de Dados:**

- **Object:** Representa uma coleção de pares chave-valor.



```
1 // Tipo de Dados - Object (Objeto): Representa uma coleção de pares chave-valor.  
2 let pessoa = {  
3     nome: "Jonatan",  
4     idade: 29,  
5     casado: true  
6 };
```

JAVASCRIPT

- **Tipos de Dados:**

- **Array:** Representa uma coleção ordenada de valores.



```
1 // Tipo de Dados - Array: Representa uma coleção ordenada de valores.  
2 let frutas = ["maçã", "banana", "laranja"];  
3 let professor = ["Jonatan", 29, true];
```

JAVASCRIPT

- **Tipos de Dados:**

- **Function:** Representa um bloco de código reutilizável.



```
1 // Tipo de Dados - Function: Representa um bloco de código reutilizável.  
2 function saudacao(nome) {  
3     return "Olá, " + nome + "!";  
4 }
```

JAVASCRIPT

- **Tipos de Dados:**

- **Date:** Representa uma data e hora.



```
1 // Date: Representa uma data e hora.  
2 let hoje = new Date();
```

JAVASCRIPT

- **Estrutura Condicional:**

- A estrutura condicional é usada para controlar o fluxo do programa com base em condições.
- As principais instruções de controle condicional são if, else, else if, switch, e as expressões condicionais ternárias.

JAVASCRIPT

- **Estrutura Condicional:**

- **If:** A estrutura de controle If é utilizada para realizar testes condicionais e tomar decisões no código com base em uma condição específica. O bloco de código dentro do if é executado apenas se a condição fornecida for verdadeira.

JAVASCRIPT

- Estrutura Condicional:
 - If:

```
1 // If
2 // A estrutura de controle If é utilizada para realizar testes condicionais
3 // e tomar decisões no código com base em uma condição específica.
4 // O bloco de código dentro do if é executado apenas se a condição fornecida for verdadeira.
5 let idadeUsuario = 30;
6
7 idadeUsuario = prompt("Digite a idade do Usuário: ");
8
9 if (idadeUsuario == 19) {
10     console.log("A idade do Usuário é igual a 19 anos.");
11 }
12
13 if (idadeUsuario > 25) {
14     console.log("A idade do Usuário é maior que 25 anos.");
15 }
```

JAVASCRIPT

- **Estrutura Condicional:**

- **If e Else:** A estrutura de controle If e Else é utilizada para tomar decisões com base em uma condição. Ela permite a execução de diferentes blocos de código, dependendo se a condição é verdadeira ou falsa.

JAVASCRIPT

- Estrutura Condicional:
 - If e Else:

```
1 // If e Else
2 // A estrutura de controle If e Else é utilizada para tomar decisões com base em uma condição.
3 // Ela permite a execução de diferentes blocos de código,
4 // dependendo se a condição é verdadeira ou falsa.
5 let idadePessoa = 18;
6
7 idadePessoa = prompt("Qual a sua idade? ");
8
9 if (idadePessoa >= 18) {
10     console.log("Você é maior de idade.");
11 }
12 else {
13     console.log("Você é menor de idade.");
14 }
```

JAVASCRIPT

- **Estrutura Condicional:**

- **If, Else If e Else:** A estrutura If, Else If e Else é usada para tomar decisões com base em condições múltiplas. Ela permite avaliar várias condições sequencialmente até que uma delas seja verdadeira, executando o bloco de código associado a essa condição. Se nenhuma condição for verdadeira e houver um bloco else, o código dentro dele será executado.

JAVASCRIPT

- Estrutura Condicional:
 - If, Else If e Else:

```
1 // If, Else If e Else
2 // A estrutura If, Else If e Else é usada para tomar decisões com base em condições múltiplas.
3 // Ela permite avaliar várias condições sequencialmente até que uma delas seja verdadeira,
4 // executando o bloco de código associado a essa condição.
5 // Se nenhuma condição for verdadeira e houver um bloco else, o código dentro dele será executado.
6 let notaAluno = 50;
7
8 notaAluno = prompt("Informe a nota do aluno: ");
9
10 if (notaAluno >= 90) {
11     console.log("A nota do aluno é A.");
12 }
13 else if (notaAluno >= 80) {
14     console.log("A nota do aluno é B.");
15 }
16 else if (notaAluno >= 70) {
17     console.log("A nota do aluno é C.");
18 }
19 else {
20     console.log("A nota do aluno é D.");
21 }
```

JAVASCRIPT

- **Estrutura Condicional:**

- **Switch:** A instrução Switch é uma estrutura de controle de fluxo que permite avaliar uma expressão contra vários casos possíveis. Ela é uma alternativa mais limpa e organizada em comparação com uma série de instruções If, Else If e Else aninhadas quando há várias condições a serem verificadas.

JAVASCRIPT

- Estrutura Condicional:
 - Switch:

```
1 // Switch
2 // A instrução Switch é uma estrutura de controle de fluxo que permite
3 // avaliar uma expressão contra vários casos possíveis.
4 // Ela é uma alternativa mais limpa e organizada em comparação com uma série de
5 // instruções If, Else If e Else aninhadas quando há várias condições a serem verificadas.
6 let diaSemana = 3;
7 switch (diaSemana) {
8   case 1:
9     console.log("Domingo");
10    break;
11   case 2:
12     console.log("Segunda-feira");
13    break;
14   case 3:
15     console.log("Terça-feira");
16    break;
17   case 4:
18     console.log("Quarta-feira");
19    break;
20   case 5:
21     console.log("Quinta-feira");
22    break;
23   case 6:
24     console.log("Sexta-feira");
25    break;
26   case 7:
27     console.log("Sábado");
28    break;
29   default:
30     console.log("Dia da Semana Inválido!");
31 }
```

JAVASCRIPT

- **Estrutura Condicional:**

- **Expressão condicional ternária:** A expressão condicional ternária, também conhecida como operador ternário, é uma forma compacta e concisa de escrever uma instrução If e Else em uma única linha. Ela é frequentemente usada para atribuir um valor a uma variável com base em uma condição.

JAVASCRIPT

- **Estrutura Condicional:**
 - **Expressão condicional ternária:**

```
1 // Expressão condicional ternária
2 // A expressão condicional ternária, também conhecida como operador ternário,
3 // é uma forma compacta e concisa de escrever uma instrução If e Else em uma única linha.
4 // Ela é frequentemente usada para atribuir um valor a uma variável com base em uma condição.
5 // Sintaxe: condicao ? valorSeVerdadeiro : valorSeFalso;
6 // Se a condição for verdadeira, o valor antes dos dois pontos (:) é retornado;
7 // caso contrário, o valor depois dos dois pontos é retornado.
8 let idade = 20;
9 let status = (idade >= 18) ? 'Maior de idade' : 'Menor de idade';
10 console.log(status);
```

JAVASCRIPT

- **Estrutura de Repetição:**

- As estruturas de repetição, também conhecidas como loops, são usadas para executar um bloco de código repetidamente enquanto uma condição específica for verdadeira.
- Os principais tipos de estruturas de repetição são: while, do while, for.

JAVASCRIPT

- **Estrutura de Repetição:**

- **While:** A estrutura de repetição while executa um bloco de código enquanto a condição fornecida for verdadeira. A condição é avaliada antes de cada iteração (repetição).

JAVASCRIPT

- Estrutura de Repetição:
 - While:

```
1 // While
2 // A estrutura de repetição while executa um bloco de código enquanto a condição fornecida for verdadeira.
3 // A condição é avaliada antes de cada iteração (repetição).
4 let contador = 0;
5
6 while (contador <= 10) {
7   console.log("O contador é " + contador);
8   contador = contador + 1;
9 }
```

JAVASCRIPT

- **Estrutura de Repetição:**

- **Do While:** A estrutura de repetição do while é semelhante ao while, mas a condição é avaliada após a execução do bloco de código. Isso garante que o bloco de código seja executado pelo menos uma vez, mesmo se a condição inicial for falsa.

JAVASCRIPT

- Estrutura de Repetição:
 - Do While:

```
1 // Do While
2 // A estrutura de repetição do while é semelhante ao while, mas a condição é avaliada após a execução do bloco de código.
3 // Isso garante que o bloco de código seja executado pelo menos uma vez, mesmo se a condição inicial for falsa.
4 contador = 0;
5
6 do {
7     console.log("O contador é " + contador);
8     contador = contador + 1;
9 } while (contador <= 10);
10
```

JAVASCRIPT

- **Estrutura de Repetição:**

- **For:** A estrutura de repetição for é usada quando o número de iterações é conhecido antecipadamente. Ele consiste em três partes: inicialização, condição de continuação e expressão de incremento/decremento.

JAVASCRIPT


- **Estrutura de Repetição:**

- **For:**

```
1
2 // For
3 // A estrutura de repetição for é usada quando o número de iterações é conhecido antecipadamente.
4 // Ele consiste em três partes: inicialização, condição de continuação e expressão de incremento/decremento.
5
6 for (let i = 0; i < 5; i = i + 1) {
7   console.log(i);
8 }
```


JAVASCRIPT

- **Estrutura de Repetição:**
 - **Gerenciamento de Loop:**
 - `break`: Encerra imediatamente o loop.



```
1 // break: Encerra imediatamente o loop.
2 for (let i = 0; i < 5; i = i + 1) {
3   if (i === 2) {
4     console.log("Entrou no Break!");
5     break;
6   }
7   console.log(i);
8 }
```

JAVASCRIPT

- **Estrutura de Repetição:**
 - **Gerenciamento de Loop:**
 - `continue`: Pula para a próxima iteração do loop.



```
1 // continue: Pula para a próxima iteração do loop.
2 for (let i = 0; i < 5; i = i + 1) {
3   if (i === 2) {
4     console.log("Entrou no Continue!");
5     continue;
6   }
7   console.log(i);
8 }
```

JAVASCRIPT

■ Operadores Aritméticos:

- Os operadores aritméticos são utilizados para realizar operações matemáticas em variáveis numéricas.
- Os operadores aritméticos básicos são: Adição (+), Subtração (-), Multiplicação (*), Divisão (/), Módulo (%), Potenciação (Math.Pow), Raiz Quadrada (Math.Sqrt), Incremento (++), Decremento (--).

JAVASCRIPT

- **Operadores Aritméticos:**
 - **Adição (+):** Soma dois operandos.



```
1 // Adição (+): Soma dois operandos.  
2 let adicao = 5 + 3;  
3 console.log(adicao);
```

JAVASCRIPT

- **Operadores Aritméticos:**

- **Subtração (-):** Subtrai o operando da direita do operando da esquerda.



```
1 // Subtração (-): Subtrai o operando da direita do operando da esquerda.  
2 let subtracao = 10 - 3;  
3 console.log(subtracao);
```

JAVASCRIPT

- **Operadores Aritméticos:**
 - **Multiplicação (*):** Multiplica dois operandos.



```
1 // Multiplicação (*): Multiplica dois operandos.  
2 let multiplicacao = 4 * 2;  
3 console.log(multiplicacao);
```

JAVASCRIPT

- **Operadores Aritméticos:**

- **Divisão (/):** Divide o operando da esquerda pelo operando da direita.



```
1 // Divisão (/): Divide o operando da esquerda pelo operando da direita.  
2 let divisao = 8 / 2;  
3 console.log(divisao);
```

JAVASCRIPT

- **Operadores Aritméticos:**


- **Módulo (%):** Retorna o resto da divisão do operando da esquerda pelo operando da direita.



```
1 // Módulo (%): Retorna o resto da divisão do operando da esquerda pelo operando da direita.  
2 let resto = 10 % 3;  
3 console.log(resto);
```


JAVASCRIPT

- **Operadores Aritméticos:**
 - **Incremento (++):** Adiciona 1 ao operando.



```
1 // Incremento (++): Adiciona 1 ao operando.
2 for (let i = 0; i < 10; i++) {
3   console.log("O i é " + i);
4 }
```

JAVASCRIPT

- **Operadores Aritméticos:**
 - **Decremento (--):** Subtrai 1 do operando.



```
1 // Decremento (--): Subtrai 1 do operando.  
2 for (let i = 10; i > 0; i--) {  
3   console.log("O i é " + i);  
4 }
```

JAVASCRIPT

▪ Operadores Lógicos:

- Os operadores lógicos são utilizados para realizar operações de lógica booleana em expressões condicionais.
- Os operadores lógicos padrão são:
 - E lógico (&&)
 - OU lógico (||)
 - NÃO lógico (!)

JAVASCRIPT

- **Operadores Lógicos:**

- **E lógico (&&):** Retorna verdadeiro se ambos os operandos forem verdadeiros.



```
1  //E Lógico (&&): Retorna verdadeiro se ambos os operandos forem verdadeiros.  
2  let x = true;  
3  let y = false;  
4  let resultado = x && y;
```

JAVASCRIPT

- **Operadores Lógicos:**

- **OU Lógico (||):** Retorna verdadeiro se pelo menos um dos operandos for verdadeiro.



```
1 //OU Lógico (||): Retorna verdadeiro se pelo menos um dos operandos for verdadeiro.  
2 let x = true;  
3 let y = false;  
4 let resultado = x || y;
```

JAVASCRIPT

- **Operadores Lógicos:**

- **NÃO Lógico (!):** Retorna o valor oposto, inverte o valor de verdadeiro para falso e vice-versa.



```
1 //NÃO Lógico (!): Inverte o valor de verdadeiro para falso e vice-versa.  
2 let x = true;  
3 let resultado = !x;
```

JAVASCRIPT

■ Operadores Relacionais:

- Os operadores relacionais são utilizados para comparar valores e determinar a relação entre eles. Eles retornam um valor booleano (verdadeiro ou falso) que indica se a relação entre os operandos é verdadeira ou falsa.
- Os principais operadores relacionais em JavaScript: Igual a (==), Estritamente Igual a (===), Diferente de (!=), Estritamente Diferente de (!==), Maior que (>), Menor que (<), Maior ou Igual a (>=) e Menor ou Igual a (<=).

JAVASCRIPT

▪ Operadores Relacionais:


- **Igual a (==):** Verifica se dois valores são iguais, mas não leva em consideração o tipo de dado.

```
1 // Igual a (==): Verifica se dois valores são iguais, mas não leva em consideração o tipo de dado.
2 //           Os valores são convertidos, se necessário, para o mesmo tipo antes da comparação.
3 let a = 5;
4 let b = '5';
5 let resultado = a == b;
```


JAVASCRIPT

- **Operadores Relacionais:**

- **Estritamente Igual a (===):** Verifica se dois valores são iguais e do mesmo tipo de dado.



```
1 // Estritamente Igual a (===): Verifica se dois valores são iguais e do mesmo tipo de dado.
2 let a = 5;
3 let b = '5';
4 let resultado = a === b;
```

JAVASCRIPT

■ Operadores Relacionais:

- **Diferente de (!=):** Verifica se dois valores não são iguais, novamente, sem considerar o tipo de dado.



```
1 // Diferente de (!=): Verifica se dois valores não são iguais, novamente, sem considerar o tipo de dado.  
2 let a = 5;  
3 let b = '5';  
4 let resultado = a != b;
```

JAVASCRIPT

- **Operadores Relacionais:**

- **Estritamente Diferente de (!==):** Verifica se dois valores não são iguais ou não são do mesmo tipo de dado.



```
1 // Estritamente Diferente de (!==): Verifica se dois valores não são iguais ou não são do mesmo tipo de dado.  
2 let a = 5;  
3 let b = '5';  
4 let resultado = a !== b;
```

JAVASCRIPT

▪ Operadores Relacionais:

- **Maior que (>):** Verifica se o operando da esquerda é maior que o operando da direita.



```
1 // Maior que (>): Verifica se o operando da esquerda é maior que o operando da direita.  
2 let x = 10;  
3 let y = 5;  
4 let resultado = x > y;
```

JAVASCRIPT

▪ Operadores Relacionais:

- **Menor que (<):** Verifica se o operando da esquerda é menor que o operando da direita.

```
1 // Menor que (<): Verifica se o operando da esquerda é menor que o operando da direita.  
2 let x = 10;  
3 let y = 15;  
4 let resultado = x < y;
```

JAVASCRIPT

■ Operadores Relacionais:

- **Maior ou Igual a (\geq):** Verifica se o operando da esquerda é maior ou igual ao operando da direita.

```
1 // Maior ou Igual a ( $\geq$ ): Verifica se o operando da esquerda é maior ou igual ao operando da direita.  
2 let x = 10;  
3 let y = 10;  
4 let resultado = x  $\geq$  y;
```

JAVASCRIPT

▪ Operadores Relacionais:

- **Menor ou Igual a (<=):** Verifica se o operando da esquerda é menor ou igual ao operando da direita.

```
1 // Menor ou Igual a (<=): Verifica se o operando da esquerda é menor ou igual ao operando da direita.  
2 let x = 5;  
3 let y = 10;  
4 let resultado = x <= y;
```



SENAI

ATIVIDADES – PARTE 1



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

ATIVIDADES – PARTE 1

- 1) Crie um programa que exiba no console uma mensagem digitada pelo usuário.
- 2) Crie um programa que leia o nome de uma pessoa e mostre uma mensagem de boas-vindas para ela.
- 3) Crie um programa que leia os dados de um aluno: matrícula, nome, sobrenome, CPF, sexo, data de nascimento, endereço e telefone e exiba no console.

ATIVIDADES – PARTE 1

- 4) Crie um programa que leia uma disciplina, um aluno e as quatro notas desse aluno naquela disciplina. Ao final, mostre na tela a média do aluno na disciplina.
- 5) Crie um programa que exiba na tela a tabuada do número digitado pelo usuário.
- 6) Crie um programa que leia o nome e o salário de um funcionário, mostrando no final uma mensagem.

ATIVIDADES – PARTE 1

- 7) Crie um programa que leia 5 números e mostre o somatório entre eles.
- 8) Crie um programa que leia a idade de uma pessoa expressa em anos, meses e dias e mostre-a expressa apenas em dias.
- 9) Crie um programa que leia a idade de uma pessoa expressa em dias e mostre-a expressa em anos, meses e dias.

ATIVIDADES – PARTE 1

- 10) Crie um programa que leia o tempo de duração de uma atividade em horas, minutos e segundos e mostre-o expresso em segundos.
- 11) Crie um programa que leia o tempo de duração de uma atividade em segundos e mostre-o expresso em horas, minutos e segundos.
- 12) Crie um programa que leia o preço de um produto, calcule e mostre o seu preço promocional, com 15% de desconto.

ATIVIDADES – PARTE 1

- 13) Crie um programa que leia o salário de um funcionário, calcule e mostre o seu novo salário, com 27,50% de aumento.
- 14) Crie um programa que leia o valor da matrícula de um curso, calcule e mostre o seu novo valor, com 20% de desconto.
- 15) Crie um programa que leia o valor de um boleto que será pago com atraso, a quantidade de dias em atraso, calcule e mostre o seu novo valor com uma taxa de 1,25% ao dia.



SENAI

ATIVIDADES – PARTE 2



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

ATIVIDADES – PARTE 2

- 1) Crie um programa que gera e escreve os números ímpares entre 0 e 50. No início da 30ª iteração o sistema deve sair do laço de repetição.
- 2) Crie um programa que gera e escreve os números pares entre 0 e 50. No início da 30ª iteração o sistema deve pular o laço de repetição.
- 3) Crie um programa para receber um número qualquer e exibir no console se é par ou ímpar.

ATIVIDADES – PARTE 2

- 4) Crie um programa que leia um valor inicial A e exiba a sequência de valores do cálculo de A! e o seu resultado.

Ex: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

- 5) Crie um programa que efetue a soma de todos os números ímpares que são múltiplos de três e que se encontram no conjunto dos números de 1 até 100.

ATIVIDADES – PARTE 2

- 6) Crie um programa que leia os valores A, B, C e exiba no console os números ordenados em ordem crescente e depois em ordem decrescente.
- 7) Crie um programa que leia dois valores inteiros A e B se os valores forem iguais deverá se somar os dois, caso contrário multiplique A por B. Ao final de qualquer um dos cálculos deve-se atribuir o resultado para uma variável C e mostrar seu conteúdo no console.

ATIVIDADES – PARTE 2

- 8) Crie um programa que leia os dados de um aluno: nome, matrícula, disciplina, nota. Se a nota do aluno for maior ou igual a 60 exiba no console “Aprovado”, se a nota do aluno for menor que 60 porém maior ou igual a 50 exiba no console “Em recuperação” e se a nota do aluno for menor que 50 exiba no console “Reprovado”.

ATIVIDADES – PARTE 2

- 9) Crie um programa que leia a idade de 10 pessoas, exibindo ao final do programa:
- Qual é a média de idade do grupo
 - Quantas pessoas tem mais de 20 anos
 - Quantas pessoas tem menos de 10 anos
 - Qual foi a maior idade lida
 - Qual foi a menor idade lida

ATIVIDADES – PARTE 2

10) Faça um programa que leia a largura e o comprimento de um terreno retangular, calculando e mostrando a sua área em m^2 (largura x comprimento). O programa também deve mostrar a classificação desse terreno, de acordo com a lista abaixo:

- Abaixo de $100m^2$ = TERRENO POPULAR
- Entre $100m^2$ e $500m^2$ = TERRENO MASTER
- Acima de $500m^2$ = TERRENO VIP

ATIVIDADES – PARTE 2

11) Crie um programa que calcule o que deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual a condição de pagamento escolhida e efetuar o cálculo adequado.

Código – Condição de Pagamento

- 1 – À vista em dinheiro ou cheque, recebe 10% de desconto
- 2 – À vista no cartão de crédito, recebe 15% de desconto
- 3 – Em duas vezes, preço normal de etiqueta sem juros
- 4 – Em duas vezes, preço normal de etiqueta mais juros de 10%

ATIVIDADES – PARTE 2

- 12) Tendo como dados de entrada a altura e o sexo de uma pessoa, crie um programa que calcule seu peso ideal, utilizando as seguintes fórmulas: para homens: $(72.7 * h) - 58$; para mulheres: $(62.1 * h) - 44.7$.
- 13) Tendo como dados de entrada dois valores numéricos digitados pelo usuário, crie um programa para uma calculadora com as seguintes operações: soma, subtração, multiplicação e divisão. Após a escolha da operação exiba no console o resultado.

ATIVIDADES – PARTE 2

14) Tendo como dados de entrada o peso (em quilogramas) e a altura (em metros) de uma pessoa, crie um programa que calcule o Índice de Massa Corporal (IMC) dessa pessoa.

Calcule o IMC usando a fórmula: $IMC = \text{peso} / (\text{altura} * \text{altura})$.

Classifique o IMC da seguinte forma:

- $IMC < 18,5 \text{ Kg/m}^2$: Abaixo do Peso
- $IMC \geq 18,5 \text{ Kg/m}^2$ e $< 24,9 \text{ Kg/m}^2$: Peso Ideal
- $IMC \geq 25 \text{ Kg/m}^2$ e $< 29,9 \text{ Kg/m}^2$: Excesso de Peso
- $IMC \geq 30 \text{ Kg/m}^2$: Obesidade

ATIVIDADES – PARTE 2

- 15) Crie um programa para controlar os saques de um caixa eletrônico que deve possuir algum mecanismo para decidir o número de notas de cada valor que deve ser disponibilizado para o cliente que realizou o saque.
- O caixa eletrônico trabalhará com notas de R\$ 100,00, R\$ 50,00, R\$ 20,00, R\$ 10,00, R\$ 5,00, R\$ 2,00 e R\$ 1,00. Não serão aceitos valores com casas decimais, caso o usuário informe um valor com casa decimais o sistema deverá informar que o valor solicitado para o saque é inválido, e pedir que o usuário informe um novo valor.

ATIVIDADES – PARTE 2

- Um possível critério seria o da "distribuição ótima" no sentido de que as notas de menor valor fossem distribuídas em número mínimo possível. Por exemplo, se a quantia solicitada fosse R\$ 188,00, o programa deveria indicar uma nota de R\$ 100,00, um nota de R\$ 50,00, uma nota de R\$ 20,00, uma nota de R\$ 10,00, uma nota de R\$ 5,00, um nota de R\$ 2,00 e uma nota de R\$ 1,00.
- Com base nessas informações, crie um programa que receba o valor da quantia solicitada e retorne a distribuição das notas de acordo com o critério da distribuição ótima.



FIEMG.COM.BR/SENAI



@senaimg



@senaiminas



@senaimg