



# Preparando o ambiente de desenvolvimento

## Introdução


Um ambiente de desenvolvimento integrado (IDE) é um conjunto de software que consolida as ferramentas básicas necessárias para escrever e testar software.

Os desenvolvedores usam várias ferramentas em toda a criação, construção e teste de código de software. As ferramentas de desenvolvimento geralmente incluem editores de texto, bibliotecas de código, compiladores e plataformas de teste. Sem um IDE, um desenvolvedor deve selecionar, implementar, integrar e gerenciar todas essas ferramentas separadamente.

## Objetivos da aula

- Definir o conceito de IDEs de desenvolvimento.
- Demonstrar a instalação da do VS Code.
- Explicar a instalação e o uso do nodejs e extension node.js.

## Resumo

Uma IDE reúne muitas dessas ferramentas relacionadas ao desenvolvimento como uma única estrutura, aplicativo ou serviço.  conjunto de ferramentas integrado foi projetado para simplificar o desenvolvimento de software e pode identificar e minimizar erros de codificação e erros de digitação.

Alguns IDEs são de código aberto, enquanto outros são ofertas comerciais. Um IDE pode ser um aplicativo independente ou pode fazer parte de um pacote maior.

### **Recursos comuns de ambientes de desenvolvimento integrados**

Um IDE normalmente contém um editor de código, um compilador ou interpretador e um depurador, acessado por meio de uma única interface gráfica do usuário (GUI). O usuário escreve e edita o código-fonte no editor de código. O compilador traduz o código-fonte em uma linguagem legível que é executável para um computador. E o depurador testa o software para que o desenvolvedor consiga resolver quaisquer problemas ou bugs.

Um IDE também pode conter recursos como editores programáveis, modelagem de objetos e dados, teste de unidade, uma biblioteca de código-fonte e ferramentas de automação de compilação.

A barra de ferramentas de um IDE se parece muito com a barra de ferramentas de um processador de texto. A barra de ferramentas facilita a organização baseada em cores, formatação de código-fonte, diagnóstico e relatório de erros e preenchimento de código inteligente. Por meio da interface de um IDE, um desenvolvedor ou equipe de desenvolvedores pode compilar e executar código de forma incremental e gerenciar alterações no código-fonte de maneira uniforme. Os IDEs geralmente são projetados para integração com

bibliotecas de controle de versão de terceiros, como GitHub e Subversion do Apache.



## **Benefícios de usar IDEs**

Um IDE pode melhorar a produtividade dos desenvolvedores de software graças à rápida configuração e padronização das ferramentas. Sem um IDE, os desenvolvedores gastam tempo decidindo quais ferramentas usar para várias tarefas, configurando as ferramentas e aprendendo como usá-las. Muitas ou mesmo todas as ferramentas de teste de desenvolvimento necessárias estão incluídas em um ambiente de desenvolvimento integrado.


Os IDEs também são projetados com todas as suas ferramentas em uma interface de usuário. Um IDE pode padronizar o processo de desenvolvimento organizando os recursos necessários para o desenvolvimento de software na interface do usuário.

## **Tipos de IDEs e ferramentas disponíveis**

Os desenvolvedores devem combinar o IDE que usam com o tipo de aplicativo que desejam produzir. Por exemplo, se um desenvolvedor deseja criar um aplicativo no iOS, ele precisa de um IDE que suporte a linguagem de programação Swift da Apple. Os tipos de IDEs variam, sejam baseados na Web ou baseados em nuvem a móveis.

Os IDEs baseados na Web são adequados para o desenvolvimento de aplicativos baseados na Web em HTML, JavaScript ou linguagens de programação semelhantes.

O Visual Studio Code da Microsoft é um exemplo de ferramenta com recursos como editor de código, realce de sintaxe, execução de código e depuração.

O Visual Studio Code foi a primeira ferramenta de desenvolvimento multiplataforma na família Microsoft Visual Studio que roda em  Windows, Linux e Mac OS.

Essa IDE que iremos utilizar aqui nessa disciplina.

O Visual Studio Code não é um editor de código simples, em vez disso, é um ambiente poderoso que coloca a escrita de código como principal objetivo. Isso torna mais fácil escrever código para plataformas web, móveis e em nuvem para qualquer desenvolvedor que trabalha em diferentes sistemas operacionais, como Windows, Linux e macOS, tornando-o independente de ambientes de desenvolvimento proprietários.


Para combinar adequadamente todos os recursos em uma ferramenta, o Visual Studio Code fornece um ambiente de codificação baseado em pastas, que facilita o trabalho com arquivos de código que não estão organizados em projetos e oferece uma maneira unificada de trabalhar com diferentes linguagens. Começando dessa suposição, o Code oferece uma experiência de edição avançada com recursos que são comuns a qualquer linguagem, além de alguns recursos que estão disponíveis para linguagens específicas.

## **Conteúdo Bônus**

Recomendo que você faça uma pesquisa sobre o Visual Studio Code, uma poderosa ferramenta de desenvolvimento multiplataforma e gratuita.

- Nome da obra: Visual Studio Code

- Autor: Desenvolvido pela equipe da Microsoft


- Plataforma: Você pode pesquisar sobre o Visual Studio Code no mecanismo de busca de sua preferência. 

O Visual Studio Code é um ambiente de codificação versátil que oferece uma ampla gama de recursos para desenvolvedores. Ele é projetado para tornar a escrita de código mais eficiente e produtiva, independentemente do sistema operacional em que você está trabalhando, seja Windows, Linux ou macOS.

Alguns dos recursos do Visual Studio Code incluem:

1. Editor de código poderoso com realce de sintaxe para várias linguagens de programação.
2. Execução e depuração de código diretamente no editor.
3. Ambiente de codificação baseado em pastas que facilita o trabalho com arquivos de código não organizados em projetos.
4. Suporte a extensões que adicionam funcionalidades extras ao editor.
5. Recursos avançados de edição, como refatoração, preenchimento automático de código e formatação.
6. Integração com controle de versão, como Git.

Essa ferramenta é amplamente utilizada por desenvolvedores de todo o mundo e oferece uma experiência de codificação poderosa e personalizável. Ao pesquisar sobre o Visual Studio Code, você encontrará informações detalhadas, tutoriais e recursos adicionais para aprofundar seu conhecimento.

Lembre-se de que, mesmo sendo uma ferramenta gratuita, o Visual Studio Code é uma excelente opção para o desenvolvimento  software em diversas linguagens e plataformas.

## Referência Bibliográfica

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6ª Ed. Porto Alegre: Bookman, 2013.

FREEMAN, Eric. **Use a cabeça!:** programação HTML. 1ª Ed. São Paulo: Alta Books, 2016.

## ATIVIDADE PRÁTICA

**Título da Prática:** Aplicar os conceitos iniciais de JavaScript


**Objetivos:** Compreender o input de dados

**Materiais, Métodos e Ferramentas:** Para realizar esta prática vamos utilizar o Visual Studio Code.

### Prática

Bem-vindo ao exercício de solicitar o nome completo do usuário! Neste problema, nosso objetivo é criar um programa usando o Node.js que solicite ao usuário que digite seu nome e sobrenome e, em seguida, imprima o nome completo no console.

Para resolver esse problema, vamos utilizar o módulo `readline-sync` do Node.js, que nos permite interagir com o usuário no terminal. Antes de

começar, verifique se você tem o pacote `readline-sync` instalado executando o comando `npm install readline-sync` no terminal. 

A primeira etapa consiste em importar o módulo `readline-sync` no início do nosso programa. Em seguida, usaremos a função `question` desse módulo para solicitar ao usuário que digite seu nome. Vamos armazenar o nome fornecido pelo usuário em uma variável chamada `nome`.

Depois disso, repetiremos o processo para solicitar ao usuário que digite seu sobrenome, armazenando-o em uma variável chamada `sobrenome`.

A última etapa é concatenar o nome e sobrenome fornecidos pelo usuário em uma única string, que chamaremos de `nomeCompleto`. Em seguida, usaremos a função `console.log` para imprimir o nome completo no console.

Agora que temos uma visão geral do problema e dos passos necessários para resolvê-lo, você pode começar a escrever o código. Não se esqueça de testar o programa digitando diferentes nomes e sobrenomes para garantir que ele esteja funcionando corretamente.

Boa sorte e divirta-se resolvendo o exercício!

**Resolução:**

```
```javascript
```

```
const readline = require('readline-sync');
```



```
// Solicitar ao usuário que digite o nome
```

```
const nome = readline.question('Digite seu nome: ');
```

```
// Solicitar ao usuário que digite o sobrenome
```

```
const sobrenome = readline.question('Digite seu sobrenome: ');
```

```
// Imprimir o nome completo no console
```

```
const nomeCompleto = `${nome} ${sobrenome}`;
```

```
console.log('Nome completo:', nomeCompleto);
```

```
```
```

Quando você executar este código, ele irá solicitar ao usuário que digite seu nome e sobrenome. Em seguida, ele irá concatenar os dois valores e armazená-los na variável `nomeCompleto`. Por fim, o nome completo será impresso no console.

Certifique-se de ter o pacote `readline-sync` instalado antes de executar o código. Você pode instalá-lo executando o seguinte comando no terminal:

```
```
```

```
npm install readline-sync
```

```
```
```

Espero que isso ajude!

**[Ir para exercício](#)**