

# Configurando o Ambiente de Desenvolvimento para TypeScript

Por Escola Dnc

## Introdução

Neste guia, vamos explorar como configurar um ambiente de desenvolvimento profissional para trabalhar com TypeScript. Após termos visto os conceitos básicos de tipos primitivos e objetos em TypeScript, é hora de preparar nosso ambiente para projetos mais complexos. Vamos cobrir a instalação das ferramentas necessárias, a criação de um projeto TypeScript, e como executar e compilar código TypeScript.

## Ferramentas Necessárias

Para começar a desenvolver com TypeScript de forma profissional, precisamos de algumas ferramentas essenciais:

### Visual Studio Code

O Visual Studio Code (VS Code) é um editor de texto gratuito desenvolvido pela Microsoft, altamente recomendado para desenvolvimento em TypeScript.

- **Download:** Visite [code.visualstudio.com](https://code.visualstudio.com)
- **Instalação:** Baixe e siga as instruções de instalação para seu sistema operacional (Windows, Mac, ou Linux)

### Node.js

Node.js é o runtime que nos permite executar JavaScript (e, por extensão, TypeScript compilado) no servidor.

- **Download:** Acesse [nodejs.org](https://nodejs.org)
- **Versão:** Escolha a versão LTS (Long Term Support) para maior estabilidade
- **Instalação:** Siga o processo de instalação padrão para seu sistema

### NPM (Node Package Manager)

O NPM vem incluído com a instalação do Node.js e será nosso gerenciador de pacotes principal.

**Dica:** Após a instalação, verifique as versões instaladas usando os comandos: - `node -v` (para verificar a versão do Node.js) - `npm -v` (para verificar a versão do NPM)

## Criando um Projeto TypeScript

Agora que temos as ferramentas básicas instaladas, vamos criar nosso primeiro projeto TypeScript:

1. Abra o VS Code
2. Crie uma nova pasta para seu projeto (ex: "introducao-typescript")
3. Abra o terminal integrado do VS Code (Ctrl+J ou Cmd+J)

## Inicializando o Projeto

No terminal, execute os seguintes comandos:

1. Inicialize um projeto npm:

```
npm init -y
```

2. Instale o TypeScript como dependência de desenvolvimento:

```
npm install typescript --save-dev
```

3. Crie um arquivo de configuração TypeScript:

```
npx tsc --init
```

## Configurando o TypeScript

Abra o arquivo `tsconfig.json` gerado e faça as seguintes alterações:

1. Descomente e configure o `outDir` :

```
"outDir": "./build",
```

2. Configure o `rootDir` :

```
"rootDir": "./src",
```

Essas configurações definem onde o TypeScript irá procurar seus arquivos fonte (src) e onde colocará os arquivos JavaScript compilados (build).

Agora que temos nosso ambiente configurado, vamos escrever e executar um código TypeScript simples:

- ```
const paciente = { nome: "Matheus", idade: 40 }; console.log(paciente);
```

Para executar nosso código TypeScript, precisamos primeiro compilá-lo para JavaScript:

- ```
npx tsc
```

- ```
node build/index.js
```

Você verá o objeto `paciente` impresso no console.

## Dicas e Boas Práticas

- **Compilação Automática:** Para evitar compilar manualmente a cada alteração, você pode usar o modo de observação do TypeScript:

```
npx tsc --watch
```

- **Scripts NPM:** Adicione scripts úteis ao seu `package.json` :

```
"scripts": {  "build": "tsc",  "start": "node build/index.js",  "dev": "tsc --watch"}
```

- **Extensões VS Code:** Instale extensões úteis como "ESLint" e "Prettier" para melhorar sua experiência de desenvolvimento

## Conclusão

Neste guia, configuramos um ambiente de desenvolvimento profissional para TypeScript. Instalamos as ferramentas necessárias, criamos um projeto básico, e aprendemos como compilar e executar código TypeScript. Com este ambiente configurado, você está pronto para explorar recursos mais avançados do TypeScript e desenvolver aplicações mais complexas.

Lembre-se de que a prática é fundamental para se tornar proficiente em TypeScript. Experimente diferentes configurações, explore a documentação oficial, e não hesite em buscar recursos adicionais para aprofundar seu conhecimento.

Boa codificação!



