### **02.1- Quantos arquivos há dentro da pasta "proj\_crud\_ts"?**

Para verificar o número de arquivos dentro da pasta, você pode usar o comando ls no terminal (para sistemas Unix-like como Linux e macOS) ou dir (para Windows). No terminal, navegue até a pasta "proj\_crud\_ts" e execute:

ls | wc -l

Ou, se você estiver no Windows:

dir /a-d /b | find /c /v ""

O número retornado será o total de arquivos na pasta.

### **02.2- Estrutura Geral do Arquivo tsconfig.json**

O tsconfig.json é um arquivo de configuração para o compilador TypeScript. Ele define como o TypeScript deve compilar os arquivos .ts no projeto. A estrutura geral do tsconfig.json é um JSON que pode conter várias propriedades. Comentários são geralmente indicados por // para uma única linha ou /\* ... \*/ para múltiplas linhas. No entanto, o formato JSON não suporta comentários diretamente, então muitas vezes os comentários são removidos antes da publicação final.

### **02.3- Função do Parâmetro compilerOptions**

O parâmetro compilerOptions é um objeto dentro do tsconfig.json que permite configurar várias opções do compilador TypeScript, como quais versões de JavaScript gerar, que módulos usar, e várias outras opções de configuração que afetam o processo de compilação.

### **02.4- Função do Parâmetro target**

O parâmetro target define a versão do JavaScript que o TypeScript deve gerar. Por exemplo, "target": "es5" significa que o código TypeScript será compilado para a versão ECMAScript 5.

### **02.5- Função do Parâmetro module**

O parâmetro module especifica o sistema de módulos que o TypeScript usará para gerar o código JavaScript. Exemplos incluem "commonjs", "amd", "es6", entre outros.

### **02.6- Função do Parâmetro esModuleInterop**

O parâmetro esModuleInterop facilita a interoperabilidade entre módulos CommonJS e ES6. Quando ativado, ele permite importar módulos CommonJS usando a sintaxe de importação ES6 e vice-versa.

### **02.7- Função do Parâmetro forceConsistentCasingInFileNames**

O parâmetro forceConsistentCasingInFileNames garante que a capitalização dos nomes dos arquivos seja consistente. Isso evita problemas de inconsistência que podem surgir ao usar diferentes sistemas de arquivos com diferentes regras de capitalização.

### **02.8- Função do Parâmetro strict**

O parâmetro strict ativa um conjunto de verificações rigorosas de tipo no TypeScript. Ele é uma forma conveniente de habilitar todas as opções de verificação rigorosa de uma vez. Se ele estiver comentado, não estará ativado. Caso contrário, estará ativado.

### **02.9- Função do Parâmetro skipLibCheck**

O parâmetro skipLibCheck faz com que o compilador ignore a verificação de tipos em arquivos de declaração de bibliotecas (.d.ts). Isso pode acelerar a compilação e evitar erros relacionados a definições de tipos de bibliotecas externas.

### **02.10- Função do Parâmetro include**

O parâmetro include especifica quais arquivos ou diretórios devem ser incluídos na compilação. Se estiver presente, define uma lista de padrões de arquivos ou diretórios que o TypeScript deve considerar. Caso esteja comentado, não será considerado pelo compilador.

### **02.11- Função do Parâmetro outDir**

O parâmetro outDir define o diretório de saída onde os arquivos JavaScript compilados serão colocados. Se estiver comentado, o TypeScript usará o diretório padrão.

### **02.12- Incluindo e Ativando os Parâmetros include e outDir**

Para incluir e ativar os parâmetros include e outDir, edite o arquivo tsconfig.json e adicione o seguinte:

{  
 "compilerOptions": {  
 "outDir": "./build"  
 },  
 "include": ["src"]  
}

Isso diz ao TypeScript para colocar os arquivos compilados no diretório ./build e para incluir todos os arquivos dentro da pasta src.

### **02.13- O que é "src" e "./build"**

* "src" é um diretório de origem onde você coloca seus arquivos TypeScript.
* "./build" é o diretório onde os arquivos JavaScript compilados serão gerados.

### **02.14- Estrutura de Diretórios na Pasta "proj\_crud\_ts"**

Com a configuração dada, a estrutura de diretórios ficaria assim:

proj\_crud\_ts  
│  
├── src # Diretório de origem onde os arquivos TypeScript estão localizados  
│ └── ... # Outros arquivos TypeScript  
│  
└── build # Diretório de saída onde os arquivos JavaScript compilados serão gerados  
 └── ... # Outros arquivos JavaScript gerados pela compilação