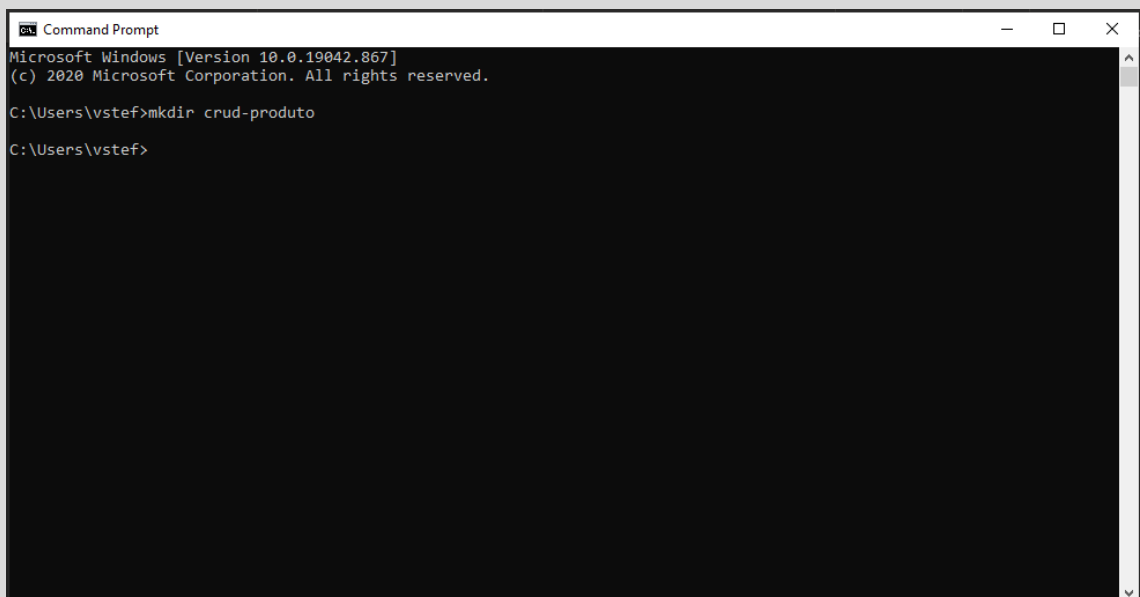


Criando o projeto no Angular

1. Abra o CMD do Windows (no menu, pesquise por “cmd”) e digite o comando **mkdir crud-projeto**; ele permitirá que você crie uma pasta no sistema operacional.



```
Command Prompt
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\vstef>mkdir crud-produto

C:\Users\vstef>
```

Importante



É de extrema importância que você se atente ao caminho utilizado para a execução dos comandos. No exemplo, o caminho é **C:\Users\vstef\crud-produto**.

2. Feito isso, crie o projeto no Angular. Acesse a pasta que você criou com o comando **cd crud-produto**; caso tenha criado sua pasta com outro nome, utilize o comando “**cd nome-da-sua-pasta**”. O comando **cd** entra nas pastas; se quiser voltar, utilize o comando “**cd ..**”.

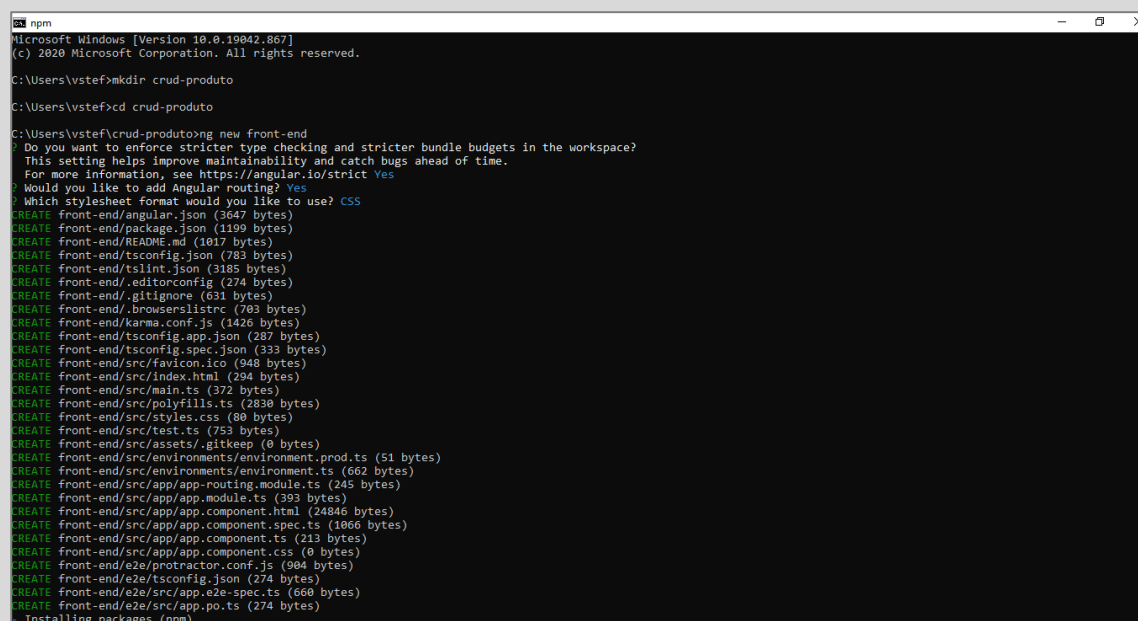
```
Command Prompt
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\vstef>mkdir crud-produto
C:\Users\vstef>cd crud-produto
C:\Users\vstef\crud-produto>
```

3. Agora, digite o comando **ng new front-end**.

Surgirão algumas perguntas sobre a configuração do projeto. Digite **y** (de “yes”) para todas e, para a estilização, escolha a opção **CSS**. Caso deseje, pode escolher outra, porém, neste curso, utilizaremos como padrão o método de estilização do CSS.

A criação do projeto pode demorar alguns minutos.



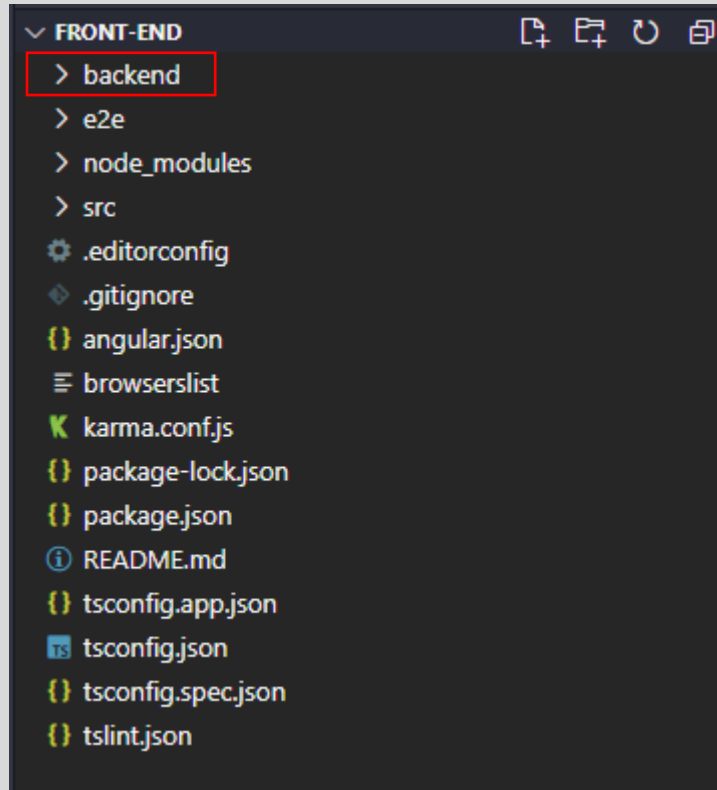
```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\vstef>mkdir crud-produto
C:\Users\vstef>cd crud-produto
C:\Users\vstef\crud-produto>ng new front-end
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE front-end/angular.json (3647 bytes)
CREATE front-end/package.json (1199 bytes)
CREATE front-end/README.md (1017 bytes)
CREATE front-end/tsconfig.json (783 bytes)
CREATE front-end/tslint.json (3185 bytes)
CREATE front-end/.editorconfig (274 bytes)
CREATE front-end/.gitignore (631 bytes)
CREATE front-end/.browserslistrc (703 bytes)
CREATE front-end/karma.conf.js (1426 bytes)
CREATE front-end/tsconfig.app.json (287 bytes)
CREATE front-end/tsconfig.spec.json (333 bytes)
CREATE front-end/src/favicon.ico (948 bytes)
CREATE front-end/src/index.html (294 bytes)
CREATE front-end/src/main.ts (372 bytes)
CREATE front-end/src/polyfills.ts (2830 bytes)
CREATE front-end/src/styles.css (80 bytes)
CREATE front-end/src/test.ts (753 bytes)
CREATE front-end/src/assets/.gitkeep (0 bytes)
CREATE front-end/src/environments/environment.prod.ts (51 bytes)
CREATE front-end/src/environments/environment.ts (662 bytes)
CREATE front-end/src/app/app-routing.module.ts (245 bytes)
CREATE front-end/src/app/app.module.ts (393 bytes)
CREATE front-end/src/app/app.component.html (24846 bytes)
CREATE front-end/src/app/app.component.spec.ts (1066 bytes)
CREATE front-end/src/app/app.component.ts (213 bytes)
CREATE front-end/src/app/app.component.css (0 bytes)
CREATE front-end/e2e/protractor.conf.js (904 bytes)
CREATE front-end/e2e/tsconfig.json (274 bytes)
CREATE front-end/e2e/src/app.e2e-spec.ts (660 bytes)
CREATE front-end/e2e/src/app.po.ts (274 bytes)
Installing packages (npm)...
```

4. Agora que o projeto foi criado, abra-o no **Visual Code**.

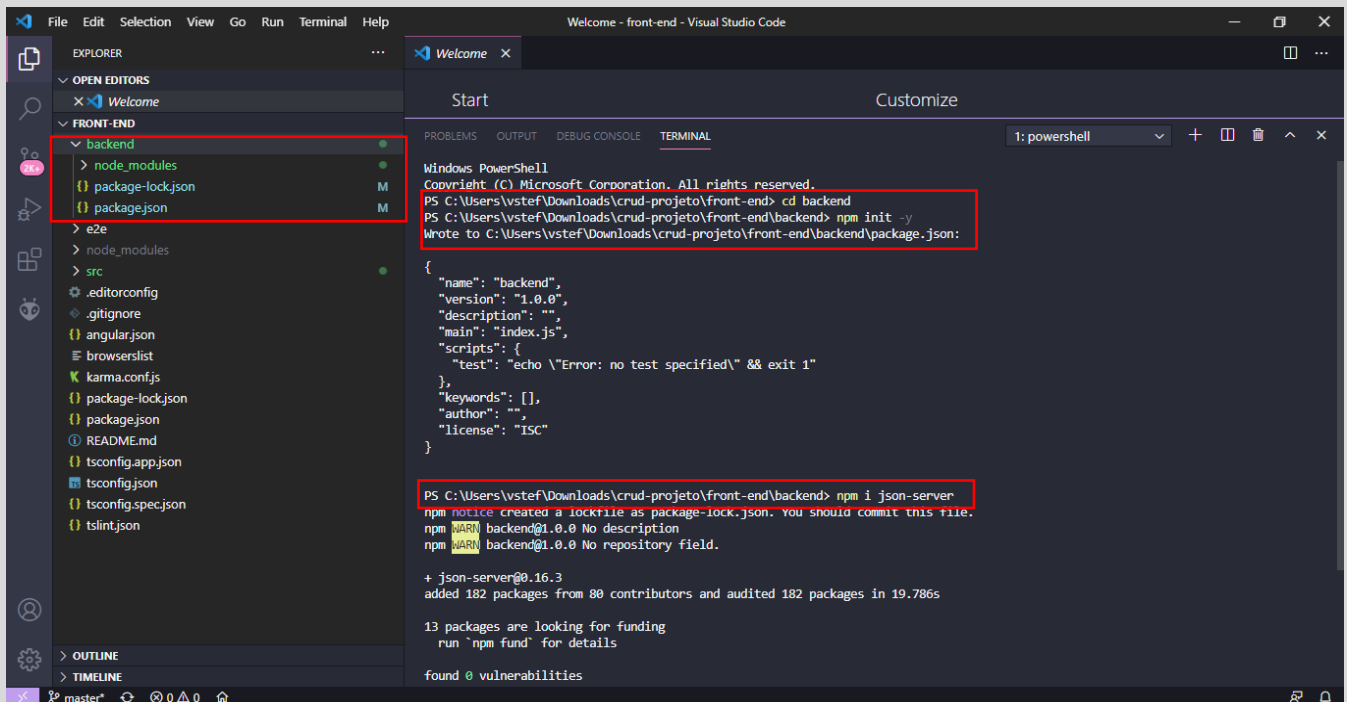
5. Para abrir um projeto, clique em **“file”**, depois, em **“open folder”** e selecione a pasta que você criou. Lembre-se de que a pasta foi criada em **C:\Users\vstef\crud-produto**. Observe o caminho pelo seu terminal (CMD).

6. Clique na pasta “Front-End” e crie uma pasta com o nome de “**backend**” (conforme destaque na imagem abaixo). Para isso, clique com o botão direito do mouse e selecione a opção “*new folder*”.



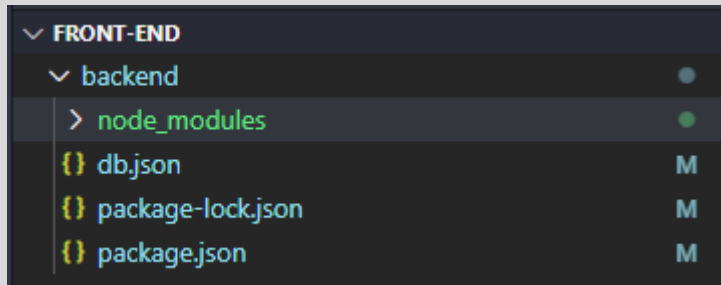
7. Agora, clique no terminal do **Visual Code**, ou digite **ctrl + shift + `** para abrir pelo atalho, acesse a pasta **backend** (`cd backend`), recém criada, e digite o comando **npm init -y**. Concluído esse procedimento, digite o comando **npm i json-server**.

Observe que serão criados dois arquivos (destacados à esquerda).

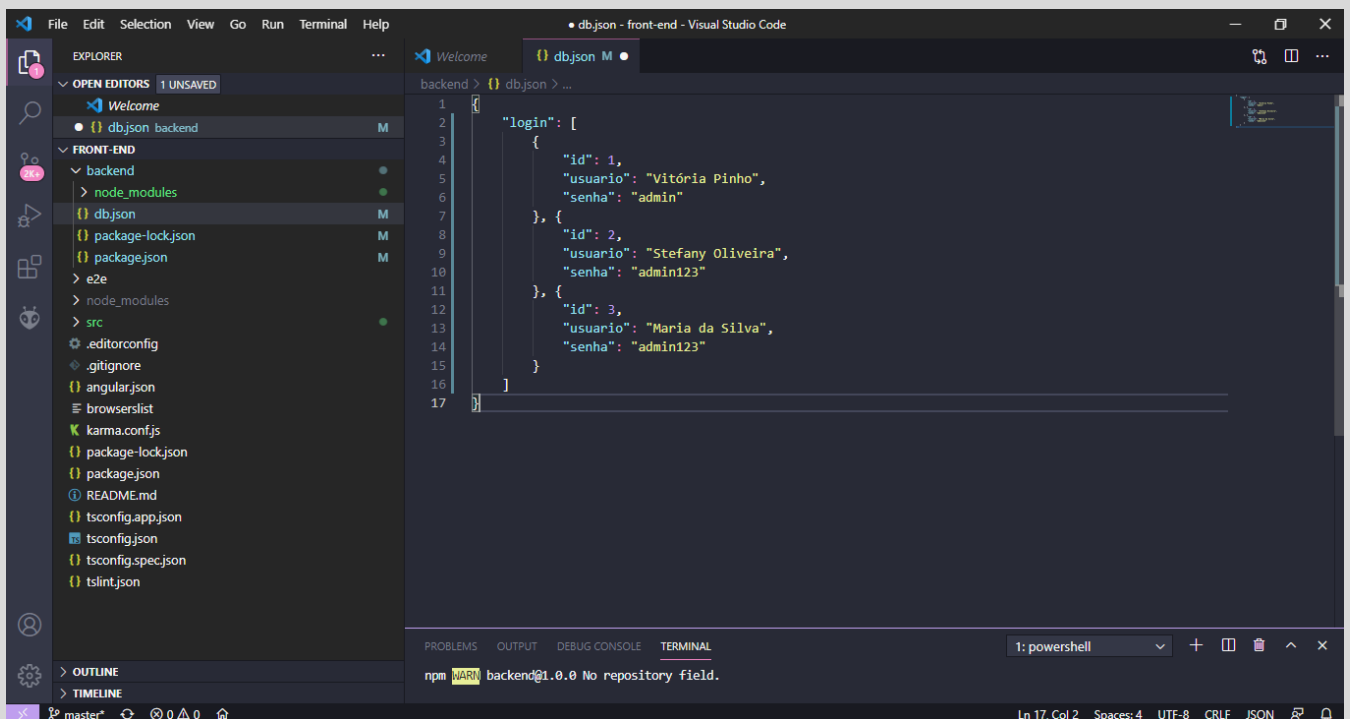


Observação: No exemplo, o caminho do terminal está diferente, pois a pasta que estava em `C:\Users\vstef\crud-produto` foi alterada para `C:\Users\vstef\Downloads\crud-produto`.

8. Agora, clique com o botão direito do mouse em **backend** e crie um arquivo (opção “*new file*”) chamado **db.json**; nessa pasta, criaremos os dados da API, que serão consumidos com o Front-End e testados no Postman.



9. Abra o arquivo db.json.



Em JSON, o conteúdo de um objeto é delimitado por chaves. No exemplo, o objeto começa com chave, seguido pelo nome “login” (entre aspas), cujo valor é um *array* de 3 objetos (delimitados por colchetes). Dentro dos colchetes, esses três objetos são delimitados por chaves e compostos de 3 pares de dados. Você pode acrescentar outros pares de dados, por exemplo:

```
{
  "id": 1,
  "nome": "Vitória",
  "e-mail": example@example.com,
  "telephone": "999999999",
  "senha": "admin123"
}
```

Lembre-se de alguns detalhes importantes: **string** (texto) deve vir entre aspas duplas. Além disso, não é necessário passar os valores em ordem crescente, mas essa é uma boa prática (recomendável, portanto). Por fim, ao adicionar mais objetos, é necessário usar a vírgula depois de fechar a chave, para marcar o final do objeto, e após um par de dados (nome/valor), quando houver outro par na sequência.

Você sabia?

De modo resumido, o objeto é um conjunto de dados. Em programação orientada a objetos, um objeto é designado para se referir a um conjunto de variáveis, funções e estrutura de dados.



Agora que os valores foram adicionados ao arquivo db.json, salve as alterações (ctrl + s).