

Instruções para a entrega: mostrar para o professor na aula do dia 06/nov.

Objetivo: Fazer uma aplicação Angular e Node.js para persistir os dados no SGBD PostgreSQL ou SQLite. A Figura 1 mostra a interface da aplicação. A Figura 2 mostra o esquema da aplicação, a parte cliente será desenvolvida no Angular e estará rodando na porta 4200 (domínio localhost:4200) do servidor Node. A parte servidor fará a conexão com o SGBD PostgreSQL ou SQLite e estará rodando no servidor Node nas portas 3100 (para conectar com o SGBD PostgreSQL) ou 3101 (para conectar com o BD do SQLite). Portanto, a aplicação Angular estará numa porta/domínio diferente do servidor.

Cadastro de Alunos

ID

Nome

Idade

Salvar

Limpar

Alunos cadastrados

ID	NOME	IDADE
59	Carla Cristina	19
66	Débora Dias	39
73	Fernando Silva	33
64	João Carlos	34
53	Mariana	41

Total: 5

Figura 1 – Interface da aplicação Angular para manter um cadastro de alunos.

Observações:

- Recomenda-se fazer a atividade seguindo os passos a seguir e usando o software Visual Studio Code da Microsoft;
- Antes de começar a atividade, reproduza os passos do arquivo [Criar o servidor e BD PostgreSQL.pdf](#) ou [Criar o servidor e BD SQLite.pdf](#) para criar o servidor Node;
- O computador precisa ter instalado o software Node, para verificar digite os comandos da Figura 3. O computador também precisa ter o Angular CLI instalado (<https://www.npmjs.com/package/@angular/cli>), digite o comando a seguir no CMD do Windows para testar:

```
ng --version
```

Se ele não estiver instalado, então você precisará instalar usando o [npm](#) (node package manager):

```
npm install -g @angular/cli
```

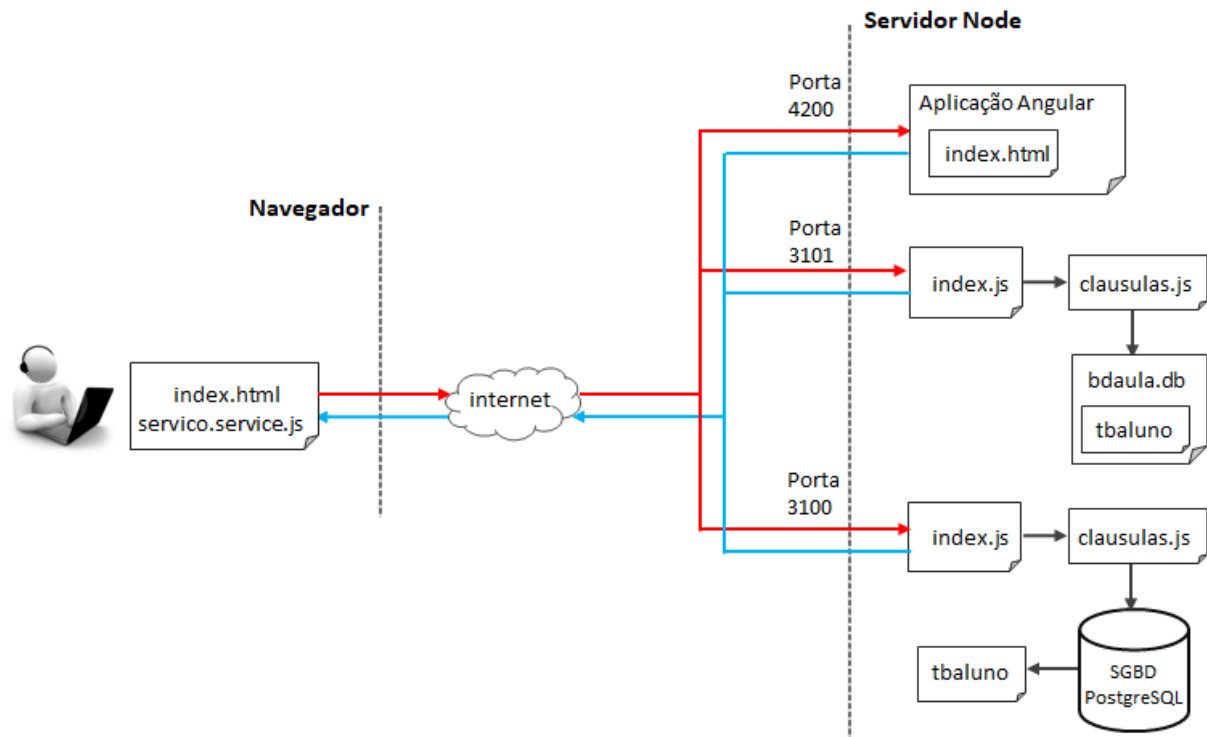


Figura 2 – Ilustração da aplicação Cliente-Servidor. Cada servidor Node roda numa porta diferente.

```

C:\> node -v
v10.15.0

C:\> npm -v
6.12.0
    
```

Figura 3 – Comandos para obter a versão do Node.js e npm.

Passo 1 – Criar o projeto Angular: acesse o CMD do Windows e vá até a pasta que você deseja criar o projeto. O comando Angular da Figura 4 cria um projeto de nome `atvnode`. A opção `--minimal` é usada para criar um projeto sem a estrutura de testes automatizados, a opção `--routing=false` indica que não queremos criar rotas para os componentes (<https://angular.io/cli/new>).

```
D:\avaliacao\Atividade_Node>ng new atvnode --minimal=true --routing=false --style=css --skipGit=true
```

Figura 4 – Comando para criar um projeto usando a Angular CLI no prompt do DOS.

Passo 2 – Subir o projeto: use o comando `ng` da Figura 5 para rodar o projeto. O projeto rodará na porta 4200 se for omitido o parâmetro `--port`. Observe que é necessário estar dentro da pasta `atvnode`.

```
D:\avaliacao\Atividade_Node\atvnode>ng serve -o --port=4201
```

Figura 5 – Comando para subir o servidor Web com a aplicação `atvnode` na porta 4201.

Passo 3 – Instalar a biblioteca Bootstrap 4: o comando da Figura 6 adiciona a biblioteca Bootstrap 4 na pasta do projeto. Observe que é necessário estar dentro da pasta `atvnode`.

Além disso, será necessário importar a biblioteca no arquivo `src/style.css`:

```
@import '~bootstrap/dist/css/bootstrap.min.css';
```

```
D:\avaliacao\Atividade_Node\atvnode>npm install bootstrap
```

Figura 6 – Adicionar a biblioteca Bootstrap 4 e suas dependências na nossa aplicação.

Passo 4 – Criar um componente no projeto: utilize o comando da Figura 7 para criar o componente que colocaremos os campos do formulário. Observe que é necessário estar dentro da pasta `atvnode`.

```
D:\avaliacao\Atividade_Node\atvnode>ng generate component form
CREATE src/app/form/form.component.html (19 bytes)
CREATE src/app/form/form.component.spec.ts (614 bytes)
CREATE src/app/form/form.component.ts (261 bytes)
CREATE src/app/form/form.component.css (0 bytes)
UPDATE src/app/app.module.ts (388 bytes)
```

Figura 7 – Comando para criar o componente de nome `form` no projeto.

Passo 5 – Criar um serviço no projeto: utilize o comando da Figura 8 para criar o serviço que colocaremos o código para fazer a conexão com o servidor Node. Observe que é necessário estar dentro da pasta `atvnode`.

```
D:\avaliacao\Atividade_Node\atvnode>ng generate service servico
CREATE src/app/servico.servico.spec.ts (338 bytes)
CREATE src/app/servico.servico.ts (136 bytes)
```

Figura 8 – Comando para criar o serviço de nome `servico` no projeto.

Passo 6 – Criar o tipo de dado Aluno: utilize o comando da Figura 9 para criar o tipo de dado Aluno e o código da Figura 10 para definir as propriedades da classe.

```
D:\avaliacao\Atividade_Node\atvnode>ng generate class aluno
CREATE src/app/aluno.spec.ts (150 bytes)
CREATE src/app/aluno.ts (23 bytes)
```

Figura 9 – Comando para criar um módulo com a classe Aluno.

```
export class Aluno {
  idaluno: number;
  nome: string;
  idade: number;
}
```

Figura 10 – Código da classe Aluno.

Passo 7 – Importações no módulo: no arquivo `src/app/app.module.ts` será necessário importar o módulo para fazer solicitações HTTP e suportar formulários com Template Driven (Figura 11).

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { AppComponent } from './app.component';
import { FormComponent } from './form/form.component';

/* necessário importar no app.module.ts */
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  declarations: [
    AppComponent,
    FormComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Figura 11 – Código do arquivo src/app/app.module.ts. Importação do módulo para fazer solicitações HTTP.

Passo 8 – Importações no módulo: substitua os arquivos do seu projeto Angular por aqueles fornecidos na pasta [src-atvnode](#) em anexo.

Passo 9 – Rodar a aplicação Angular: para rodar o projeto você precisar estar na pasta do seu projeto:

```
D:\avaliacao\atvnode\src>ng serve -o
```

Lembre-se que o servidor Node estará na porta 3100 ou 3101 e você precisará subir ele antes da aplicação Angular.

Para alternar de servidor (3100 ou 3101) altere a variável [urlbase](#) do arquivo [src/app/servico.service.ts](#):

```
urlbase: string = 'http://localhost:3100';
```