

Angular



Érico Rosiski Weber - 19/07/2024

Quem sou eu?

- Érico Rosiski Weber
- 28 anos
- Formado em Engenharia de Controle e Automação na UFSM
- Formado em Sistemas de informação na UFN
- MBA Project Management USP
- Atualmente Software Engineer
- Experiência desenvolvimento web e mobile
- Contato: ericoweber@hotmail.com



Minha experiência

- Portal Stars v1 - Web Laravel
- Apps publicados
 - Locus
 - Verden
 - Subway
 - Thaw Gear
 - CBDC Mercado Bitcoin
 - Bemis App
 - Pivotal

Demanda de novos talentos em tecnologia em **5 anos**



*Nota: TI In House tem a mesma remuneração se não maior que serviços de alto valor agregado.
Fontes: Brasscom, Bacen, IDC, Relatórios Financeiros das Estatais, RAIS e Caged, Novo Caged, Censo do Ensino Superior (INEP, 2019)

Fonte: <https://brasscom.org.br/estudo-da-brasscom-aponta-demanda-de-797-mil-profissionais-de-tecnologia-ate-2025/>



Aulas - Cronograma

14 aulas (8 web, 3 mobile, 3 tira dúvida)

- **19/07 - Angular (Web)**
- **22/07 - Angular (Web)**
- **26/07 - Angular (Web)**
- **29/07 - Angular (Web)**
- **02/08 - Angular (Web)**
- **05/08 - Angular (Web)**
- **09/08 - Angular (Web)**
- **12/08 - Angular (Web)**
- **16/08 - Flutter (Mobile)**
- **19/08 - Flutter (Mobile)**
- **23/08 - Flutter (Mobile)**
- **26/08 - Tira dúvidas**
- **30/08 - Tira dúvidas**
- **02/09 - Tira dúvidas**

Acordos

- Ninguém é dono da verdade, não saberei todas as respostas mas irei busca-las;
- É a minha primeira vez dando aula, estou muito empolgado com a oportunidade de passar a vocês um pouco do meu conhecimento;
- Amo tecnologia, ela me abriu diversas portas e me deu muitas oportunidades;
- Estou à disposição para qualquer dúvida, não tenham medo de perguntar. Se eu não souber a resposta, podem ter certeza que a buscarei para vocês;
- Ao final de toda a aula irei enviar um formulário para vocês avaliarem a aula. Será anônimo, peço que me digam o que gostaram e o que não gostaram, assim poderei melhorar a aula para vocês;

**Alguma
dúvida?**

Aula 1 - Angular

- Introdução
- Configuração e criação de projeto
- Comandos básicos
- Componentes básicos



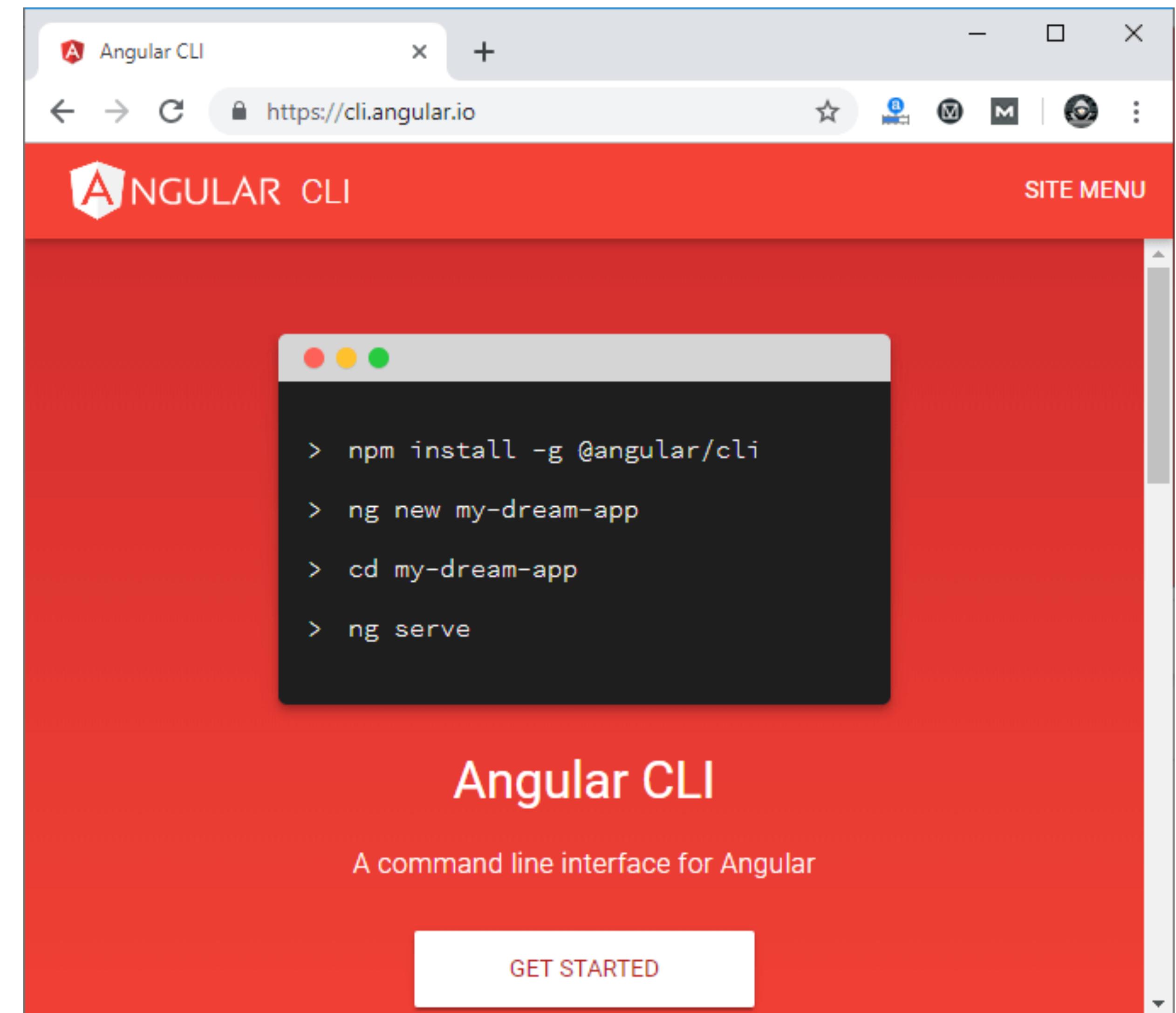
Angular

<https://angular.dev/>

- Framework para Front-End.
 - Utiliza HTML, CSS e JavaScript
 - Reativo (Data Binding Bidirecional)
 - TypeScript ("tipada", enums, compilada...)
- Criado pela Google (2016)
- Atual versão 18 (2024)
- É open source
 - Grande comunidade
 - Muito material na internet
 - Muitas empresas utilizando
- Arquitetura baseada em componentes
 - Funcionalidades isoladas e independentes
- Fornece elementos básicos:
 - Templates
 - Roteamento
 - Módulos
 - Injeção de dependência
 - Ferramentas para testes

Angular CLI

- Ferramenta oficial de comandos Angular
 - Gerar Projeto
 - Servi-lo
 - Adicionar módulos/ componentes
 - Criar testes
 - Desenvolvida em Node.js



Node.js

Tecnologia de servidor

- Permite executar códigos JS
- Lado do servidor
- Angular CLI necessita do Node.js
- Porém Angular é Front-End



<https://nodejs.org/en>

Configurando ambiente

Instalação Angular

- Primeiro passo: Instalar o Node e NPM
 - Existem muitas maneiras de se fazer isso, tais como:
 - usar o NVM (Node Version Manager) para a instalação e trabalho com muitas versões do Node em seu sistema
 - usar o gerenciador de pacotes oficial do seu sistema operacional
 - instalar a partir do site oficial da web.
 - <https://nodejs.org/en/download/>
 - Obs: Se estiver usando Windows, pref

Comandos básicos Node.js

- Verificar versão do Node instalada:

\$ node -v

- Instalar Angular CLI:

\$ npm install @angular/cli

Comandos básicos Angular CLI

- Verificar versão do Node instalada:

\$ ng version

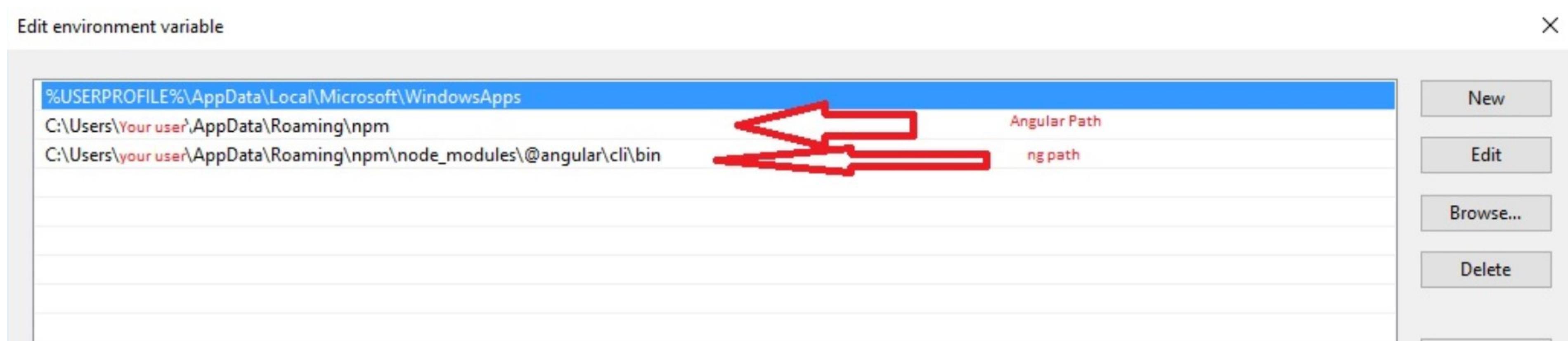
- Comando Help

\$ ng help

Extra - Solução de problemas

Se der erro, siga os seguintes passos:

- npm uninstall -g angular-cli
- npm uninstall --save-dev angular-cli
- npm install -g @angular/cli
- Depois de instalar você entra em sistema, configuração avançadas do sistema, Variáveis de Ambiente, Path, Editar
 - Se der erro no install, tente:
 - npm install -g @angular/cli --force



Comandos Angular CLI



Angular CLI

Comandos

- **add**: adiciona suporte a uma biblioteca externa em seu projeto.
- **build (b)**: compila uma aplicação do Angular em seu diretório de saída, chamado dist/ no caminho de saída fornecido. Este comando deve ser executado dentro de um diretório de espaço de trabalho.
- **config**: obtém ou configura os valores de configuração do Angular.
- **doc (d)**: abre a documentação oficial do Angular (angular.io) em um navegador, buscando uma palavra-chave determinada.
- **e2e (e)**: faz o build e serve uma aplicação do Angular, depois executa testes de ponta a ponta usando o Protractor.

Angular CLI

Comandos

- **generate (g):** gera e/ou modifica arquivos com base em um esquema.
- **help:** lista os comandos disponíveis e suas descrições breves.
- **lint (l):** executa as ferramentas de linting no código da aplicação em Angular em uma determinada pasta de projeto.
- **new (n):** cria um espaço de trabalho e uma aplicação inicial em Angular.
- **run:** executa um destino personalizado definido em seu projeto.

Angular CLI

Comandos

- **serve (s)**: faz o build e serve sua aplicação, fazendo um novo build a cada alteração de arquivos.
- **test (t)**: executa os testes unitários em um projeto.
- **update**: atualiza sua aplicação e suas dependências. Consulte <https://update.angular.io/> (em inglês)
- **version (v)**: mostra a versão da Angular CLI.
- **xi18n**: extrai as mensagens i18n do código-fonte.

Criando o primeiro projeto



Criando o primeiro projeto usando Angular CLI

\$ ng new NomeDoProjeto

Por padrão, na criação de um projeto, será perguntado se deseja adicionar alguns módulos extras:

- Would you like to add Angular routing?
 - Resposta padrão: Não. Selecione a opção SIM.
- stylesheet
 - Selecione CSS

Estrutura de diretórios projeto em Angular

Vejamos a função de cada diretório/arquivo:

- **/e2e/**: contém os testes end-to-end (simulação do comportamento do usuário) do site
- **/node_modules/**: todas as bibliotecas de terceiros são instaladas nesta pasta usando `npm install`
- **/src/**: contém o código-fonte da aplicação. A maior parte do trabalho será feita aqui
- **/app/**: contém módulos e componentes
- **/assets/**: contém os ativos estáticos, como imagens, ícones e estilos

- `/environments/`: contém arquivos de configuração específicos do ambiente (produção e desenvolvimento)
- `browserslist`: necessário para o autoprefixer para suporte ao CSS
- **`favicon.ico`: o favicon**
- **`index.html`: o arquivo HTML principal**
- `karma.conf.js`: o arquivo de configuração para o Karma (uma ferramenta de testes)
- **`main.ts`: o arquivo inicial principal, a partir de onde o AppModule é iniciado**
- `polyfills.ts`: polyfills necessários ao Angular
- **`styles.css`: o arquivo de folha de estilos (stylesheet) global do projeto**
- `slint.json`: o arquivo de configuração para o TSlint (uma ferramenta de análise estática)
- **`package.json`: contém as informações básicas do projeto (nome, descrição e dependências)**
- `README.md`: um arquivo em markdown que contém a descrição do projeto
- `tsconfig.json`: o arquivo de configuração para o TypeScript
- `tslint.json`: o arquivo de configuração para o TSlint (uma ferramenta de análise estática)

Executando a aplicação

A Angular CLI fornece um conjunto de ferramentas completo para desenvolver aplicações de front-end em sua máquina local.

Assim, não é preciso instalar um servidor para servir o projeto — você pode, simplesmente, usar o comando *ng serve* a partir do terminal para servir seu projeto localmente.

```
$ cd NomeDoProjeto  
$ ng serve
```

Componentes básicos

Criando um novo componente

\$ ng generate component meu-componente

- meu-componente é o nome do componente.
- Os componentes fornecem estrutura para organizar seu projeto em partes fáceis de entender, com responsabilidades claras, para que seu código seja sustentável e escalonável.
- A Angular CLI automaticamente adicionará uma referência a components, directives e pipes no arquivo src/app.module.ts.

Estrutura de um novo componente

- **app.component.html**: Representa a estrutura visual com diretivas do angular
- **app.component.css**: Código de estilização ou pré processador que voce escolher
- **app.component.spec.ts**: Arquivo de teste com as regras de teste para realizar testes unitários TDD
- **app.components.ts**: Arquivo typescript responsável pela lógica/regra de negócio do componente

app.components.ts

- Estrutura simples: uma classe com o nome do componente que implementa a interface onInit (do próprio angular).
- Decorator @Component
 - Selector = qual tag que vamos usar no html
 - TemplateUrl : caminho do caminho que vai ter o conteúdo visual do componente , o próprio html ja vai
 - styleUrls: caminho dos arquivos de estilização.
- Constructor = usado para injeção de dependência
- ngOnInit chamado quando todas informações

Prova Prática

Próxima aula



brincadeirinha

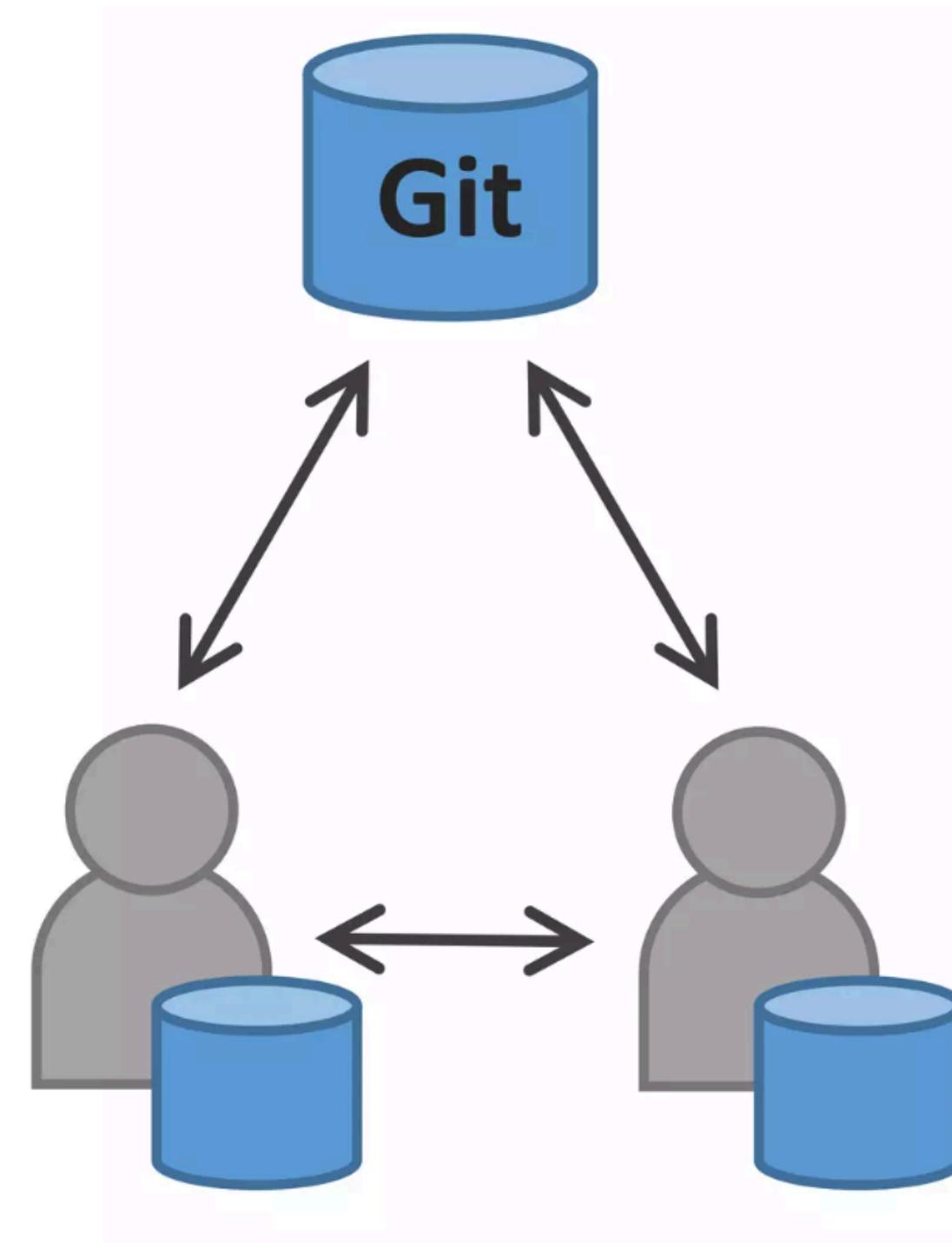
Git

Sistema de controle de versão de código

Git

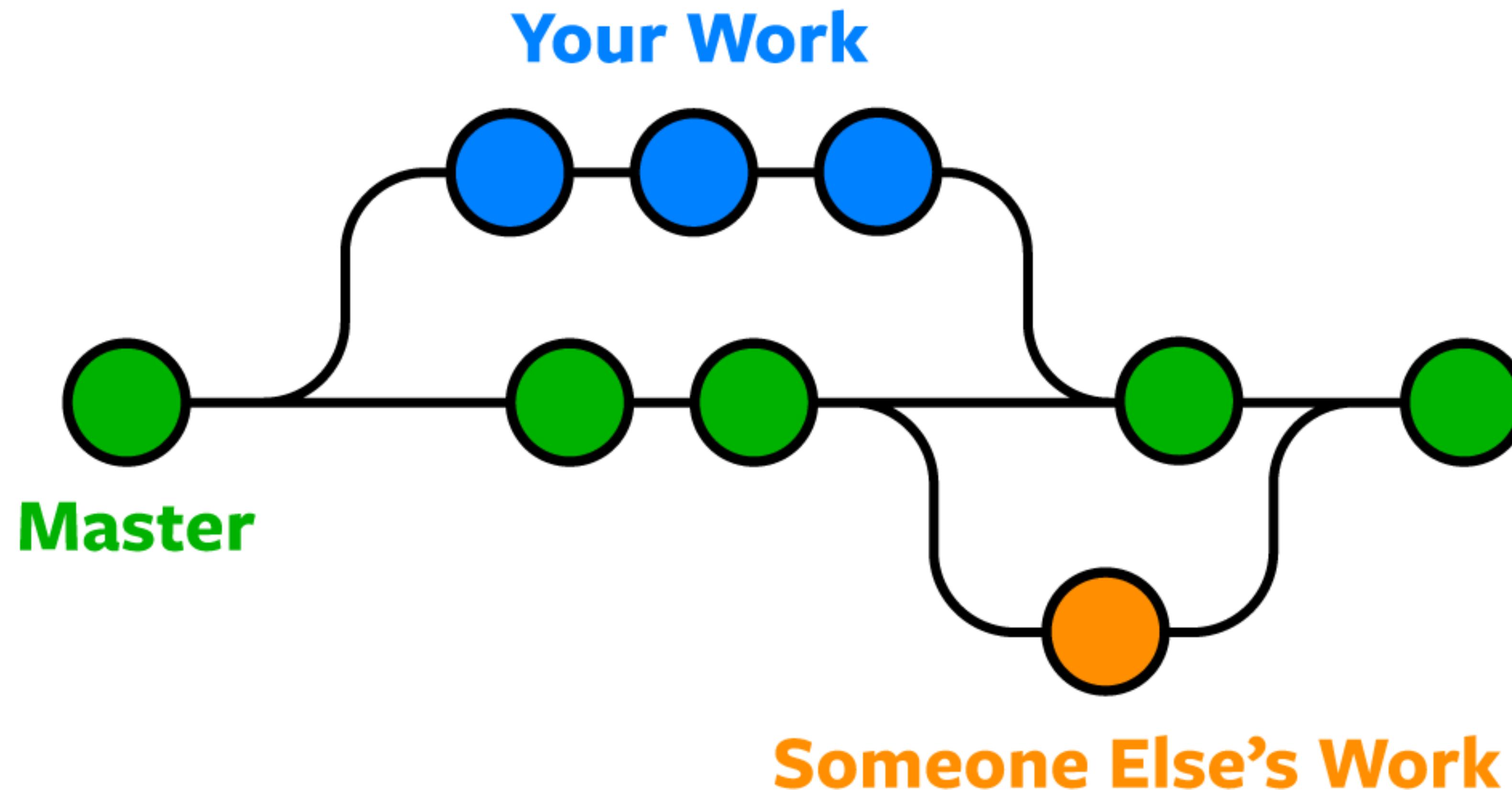
Sistema de controle de versão distribuído

- Controle de histórico
- Trabalho em equipe
- Marcação, resgate, alteração de versões
- Ramificação do projeto

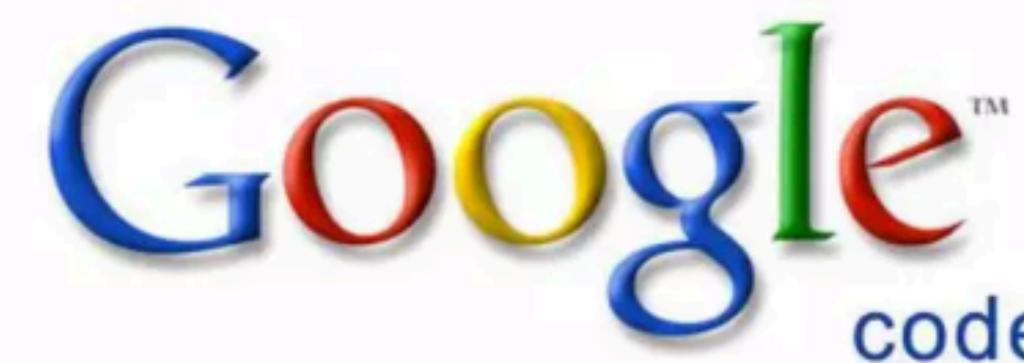


- Autonomia
Ações básicas “off-line”.
- Rapidez
Processos são locais.
- Trabalho privado
Trabalho local não afeta os demais.
- Confiabilidade
Todo repositório é um *backup*, ou seja, uma cópia completa do repositório, incluindo versões anteriores e histórico.

Git Workflow



Servidores para Hospedagem



Configurando Git no ambiente

Tutorial: Getting Started - Installing Git

Instalação do Git

Windows:

<https://git-scm.com/download/win>

Linux

```
$ sudo dnf install git-all
```

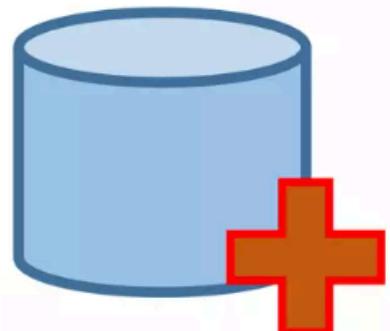
```
$ sudo apt install git-all
```

Mac OS

Ja vem instalado

Comandos básicos de Git

- Criando um repositório Git



```
$ git init
```

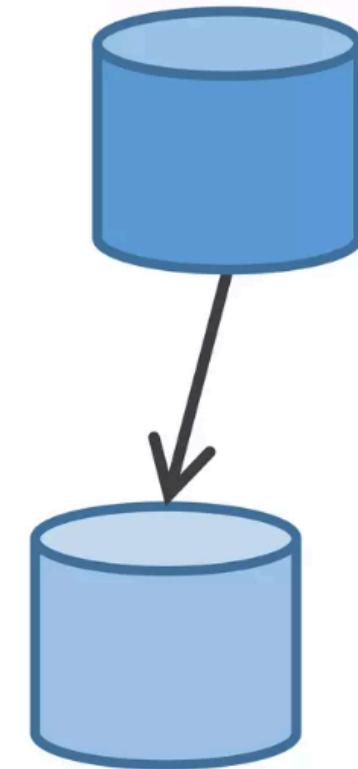
Transforma a diretório atual em um repositório git, criando o subdiretório “.git”.



```
$ git init <dir>
```

Cria o diretório <dir> e transforma em um repositório git.

- Clonando



```
$ git clone <repo>
```

Clona o repositório <repo> para a máquina local.

```
$ git clone <repo> <dir>
```

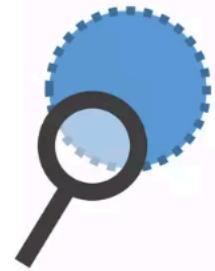
Clona o repositório <repo> para o diretório <dir>.

```
$ git clone git@github.com:user/Project.git
```

```
$ git clone https://github.com/user/Project.git
```

Comandos básicos de Git

- Rastreando mudanças locais
- Rastreando Histórico Remoto



```
$ git status
```

Lista os arquivos que estão e que não estão na área transitória, e os arquivos que não estão sendo rastreados.

```
$ git status -s
```

Lista os arquivos de uma forma simplificada.

\$ git log

Comandos básicos de Git

Fluxo simplificado para **salvar** local e remotamente as modificações

\$ git add * : adiciona as mudanças ao index provisório

\$ git commit -m “comentário” : adiciona as mudanças do index ao HEAD

\$ git push origin master : envia as alterações da HEAD local para o diretório remoto

\$ git remote add origin <servidor> : conectar seu repositório a um servidor remoto

Comandos básicos de Git

Fluxo simplificado **buscar** remotamente alterações e atualizar localmente

\$ git pull: para atualizar seu repositório local com a mais nova versão.

Remover todas as alterações e commits locais, recupere o histórico mais recente do servidor e sobrescrever local

\$ git fetch origin

\$ git reset --hard origin/master

Para próxima aula

- Ler Git - Guia Prático
- Assista os vídeos:

Introdução ao que é Angular

- Assista, pelo menos, os 4 primeiros vídeos desta playlist

Curso Introdutório Angular

Aprox. 40 min somando todos vídeos

LINK DO REPOSITORIO DO PROJETO:

<https://github.com/webererico/Angular.git>

Avalie essa aula:
Formulário avaliação Aula 1