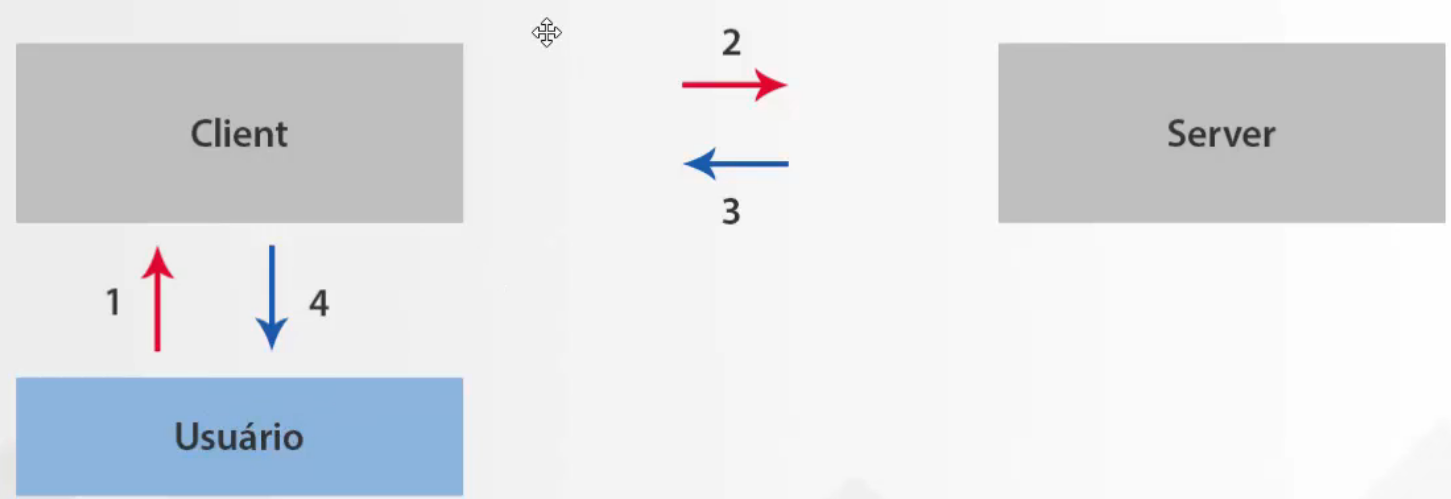
**Angular 2**

É um framework para criação de Single Page Applications (SPA).

Aplicações normais



Aplicações Single Page Application



**Sobre Angular 2**

* Rodar diretamente do servidor;
* Facilidade de SEO;
* Angular CLI;
* Mobile Toolkit;
* Material Design 2/
* Augury;
* NativeScript e Ionic 2.

**Angular CLI**

Facilita os processos necessários com o Angular 2 como componentes, serviços, diretivas e auxilia a executar rotinas de testes e implantação.

**Augury**

Esse plugin foi desenvolvido para debugar o Angular 2.

**NativeScript e Ionic 2**

NativeScript agora suporta o angular 2. É possível desenvolver uma aplicação Web e depois é convertido para uma linguagem nativo do dispositivo Mobile.

Ionic 2 desenvolvido para suportar angular 2.

**Node**

É uma plataforma construída sob o motor javascript do Google Chrome (V8). Usa o modelo orientado a eventos e tem um processo de IO não bloqueante.

**TypeScript**

É um superset tipado do Javascript que compila para Javascript plano ou seja pode escrever para os Javascripts nas versões ES3, ES4, ES5, ES6 e ES7. Ele ajuda a deixar compatível com as versões dos browsers.

O typescript (OpenSource) foi criado pela Microsoft melhor especificando (Anders Hejlsberg) o mesmo criador do C#. Uma das maiores características do typescript é a inferência de tipo (usada nas linguagens orientada a objetos). Isso ajuda em tempo de compilação verificar se o código tem erro, como também outros recursos como classes, interfaces, generics e todos os recursos do ES6.

Estrutura Angular 2

**Package.json**

Contém as dependências necessárias para executar o projeto.

* dependencies:
  + Existem 3 categorias:
    - Dependências de recursos;
      * "@angular/common": "~2.1.1":
        + São alguns recursos e serviços comum a todos.
      * "@angular/compiler": "~2.1.1":
        + Compilador do angular. Converte os templates na sintaxe do angular para ser renderizado executado.
      * "@angular/core": "~2.1.1":
        + Fornece recursos críticos para fazer a aplicação funcionar como: Decoretors, Component, ngModel, Direct e Pipe;
      * "@angular/forms": "~2.1.1":
        + Para trabalhar com formulários
      * "@angular/http": "~2.1.1":
        + Client HTTP para fazer requisições no servidor.
      * "@angular/platform-browser": "~2.1.1":
        + Incluir tudo relacionado ao DOM e ao browser. Como se fosse a tradução da arvore do dom que o angular2 utiliza. Incluir um método de compilação estática para a versão de produção (Faz uma pré compilação do código reduzindo o tamanho e aumentando o ganho de performace da aplicação). **AoT – Ahead of Time Compilation.** (Usado na faze produção)
      * "@angular/platform-browser-dynamic": "~2.1.1":
        + **JiT- Just in Time Compilation** é quando a compilação é feita no cliente. (Usado na faze desenvolvimento)
      * "@angular/router": "~3.1.1":
        + Para trabalhar com rotas.
      * "@angular/upgrade": "~2.1.1":
        + Conjunto de utilitários para atualizar aplicações angular 1.
      * "systemjs": "0.19.39":
        + Carregador de módulos. (Poderia ser substituído pelo WebPacage).
    - Dependências de Polyfills;
      * "core-js": "^2.4.1":
        + Fornece características essenciais para poder trabalhar com o ES2015 ou ES6. Quando todos os browsers passar a implementar todas as especificações do ES6 essa dependência não será mais necessária.
      * "reflect-metadata": "^0.1.8":
        + Pacote compartilhado entre o angular2 e o TypeScript. É uma dependência de aplicação de não do angular 2. Basicamente trabalha com os metadados da aplicação (Ex: Quando chamado o Decorator e passa os metadados é essa dependência que irá trabalhar com eles).
      * "rxjs": "5.0.0-beta.12":
        + Reactive Extensions for Javascript utilizado para trabalhar com o observable no angular 2.
      * "zone.js": "^0.6.25":
        + Para trabalhar com zonas no angular 2.
    - Outras (Ex: Bootstrap):
      * "angular-in-memory-web-api": "~0.1.13":
        + Simula um servidor remoto. Usado na faze de desenvolvimento.
        + Bootstrap:
        + Jquery;
* devDependencies: usadas apenas para o desenvolvimento.
  + "@types/core-js": "^0.9.34":
  + "@types/node": "^6.0.45":
  + "concurrently": "^3.0.0":
    - Para possibilitar executar comando simultâneos no sistema operacional.
  + "lite-server": "^2.2.2":
    - Servidor de arquivos estáticos que trabalhar muito bem com aplicações angular que utiliza roteamento.
  + "typescript": "^2.0.3":
    - Incluir um servidor e compilador do typescript.
    - Pacotes com o @types antes do nome são arquivos de definição para fazer o typescript entender a linguagem desses pacotes. Ex: o typescript não consegue entender o core-js o jquery. Dessa forma ele consegue entender.

Tsconfig.json

Configurações e instruções para o compilador do TypeScript.

Systemjs.config.js

É o carregador de módulos. **IIFE-Immediately-Invoked Function Expression**

**(function (global) {**

**})(this);**

**App.module.ts**

Módulo raiz da aplicação.

@NgModule({

imports: [BrowserModule]: Como a aplicação vai rodar em um browser é preciso importar.

declarations: [AppComponent]: Todos os componentes que for usar tem que declarar dentro desse Array. Se não o angular não reconhece.

bootstrap: [AppComponent]: É chamado somente no módulo root. É onde vai dizer qual o componente root da aplicação de onde vai começar a aplicação.

})

**App.component.ts**

Define uma classe e faz um importe do decorator e decora a classe como um componente. O componente recebe um seletor css.

import { Component } from '@angular/core';

@Component({

selector: 'my-app',

template: '<h1>Hello Wolrd!</h1>'

})

export class AppComponent {}

**Main.js**

É o módulo raiz é onde faz o bootstrap da aplicação.

Caso a aplicação vá para o ambiente de produção é necessário importar o platformBrowser em vez do platformBrowserDynamic.

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app.module';

const platform = platformBrowserDynamic();

platform.bootstrapModule(AppModule);