

Residência TCE/RN: Desenvolvimento de soluções Web Aula 02

Professor: José Alex

Pesquisador e Gerente de Projetos
do Smart Metropolis

Agenda

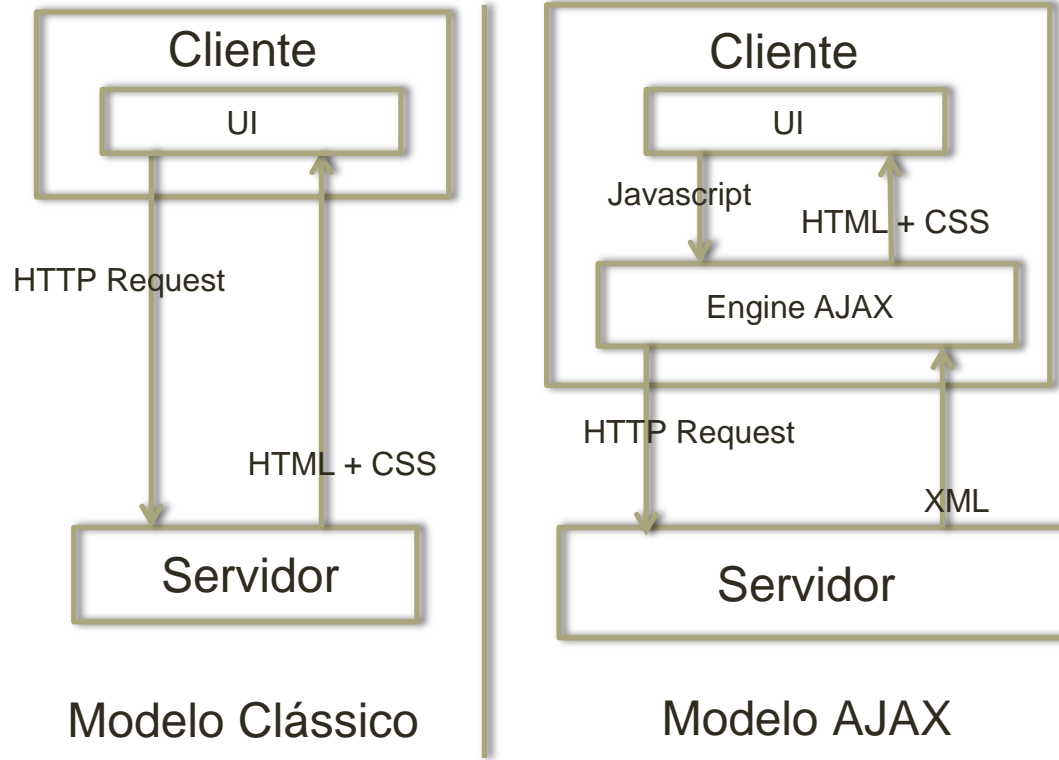
- 05/07/2019
 - [HTML 5, CSS e JavaScript](#)
- 12/07/2019
 - [Ajax, Jquery e Angular](#)
- 19/07/2019
 - [Arquitetura de serviços \(SOA\) e SOAP](#)
- 26/07/2019
 - [Serviços REST e Segurança](#)
- 02/08/2019
 - [Microserviços, MSOAM e dispositivos móveis](#)
- 09/08/2019
 - [Sistemas legados, DevOps, modelos e framework de desenvolvimento web](#)
- 16/08/2019
 - [ASP.NET Core MVC](#)
- 23/08/2019
 - [Padrões de Projeto e integração de Sistemas](#)

AJAX

AJAX

- Asynchronous Javascript and XML;
- Melhorar o nível de experiência do usuário com a aplicação web;
- Conjunto de tecnologias:
 - XHTML;
 - CSS;
 - DOM;
 - XML;
 - XSLT;
 - XMLHttpRequest;
 - Javascript.

AJAX



AJAX

- Comunicação assíncrona através do objeto XMLHttpRequest (`xmlhttp = new XMLHttpRequest();`)
- Propriedades `responseXML` ou `responseText`
- Função `Onreadystatechange`
- Propriedade `readyState` (armazena o status do `XMLHttpRequest`):
 - `0` => Requisição não iniciada;
 - `1` => Conexão com o servidor estabelecida;
 - `2` => Requisição recebida;
 - `3` => Requisição processada;
 - `4` => Requisição finalizada e resposta pronta.
- Propriedade `Status`:
 - `200` => OK;
 - `404` => Página não encontrada;

AJAX - Exemplo

```
<body>

  <div id="demo"></div>

  <form method="GET">
    <input value="Pegar a informação" type="button" onClick="getInformacao();" />
  </form>

</body>
```

```
<script language="javascript" type="text/javascript">
  function getInformacao() {
    createRequest();
    var url = "ajax.txt";
    request.open("GET", url, true);
    request.onreadystatechange = atualizaPagina;
    request.send(null);
  }
```

AJAX - Exemplo

```
<script language="javascript" type="text/javascript">
    var request = null;

    function createRequest() {
        try {
            request = new XMLHttpRequest();
        }
        catch (failed) {
            request = null;
        }

        if (request == null)
            alert("Falha ao criar o object request!");
    }
</script>
```

```
function atualizaPagina() {
    if (request.readyState == 4 && request.status == 200) {
        document.getElementById("demo").innerHTML = request.responseText;
    }
}
```


Dúvidas? Sugestões?
Comentários...

Vamos Praticar!!!

jQuery

jQuery

- Surgiu em meados dos anos 2000 em virtude de algumas limitações do Javascript na época e principalmente pelos problemas de cross-browser como uma biblioteca Javascript para trabalhar os browsers;
- Simplifica o Javascript nativo;
- Hoje está um pouco em desuso em função da evolução do javascript;
- jquery.com (versão comprimida pronta para colocar no projeto);
- Chamada externa ao arquivo;
- Pode alterar o CSS;
- Sintaxe: `$(seletor).ação(){};`
- Métodos jQuery: `$(document).ready(function()); $(seletor).click(function()); $(seletor).hide(); $(seletor).show(); $(seletor).fadeOut(); $(seletor).delay(ms); $(seletor).fadeIn(); $(seletor).css(propriedade,valor); $(seletor).text(texto); $(seletor).addClass(classe); $(seletor).removeClass(classe); $(seletor).html(texto); $.ajax();$.get(); $.post();`

jQuery - Exemplo

```
<div id="demo"></div>

<script src="jquery-3.4.1.min.js"></script>
<script>
    $.get("ajax.txt", function(resultado){
        $("#demo").html(resultado);
    })
</script>
```

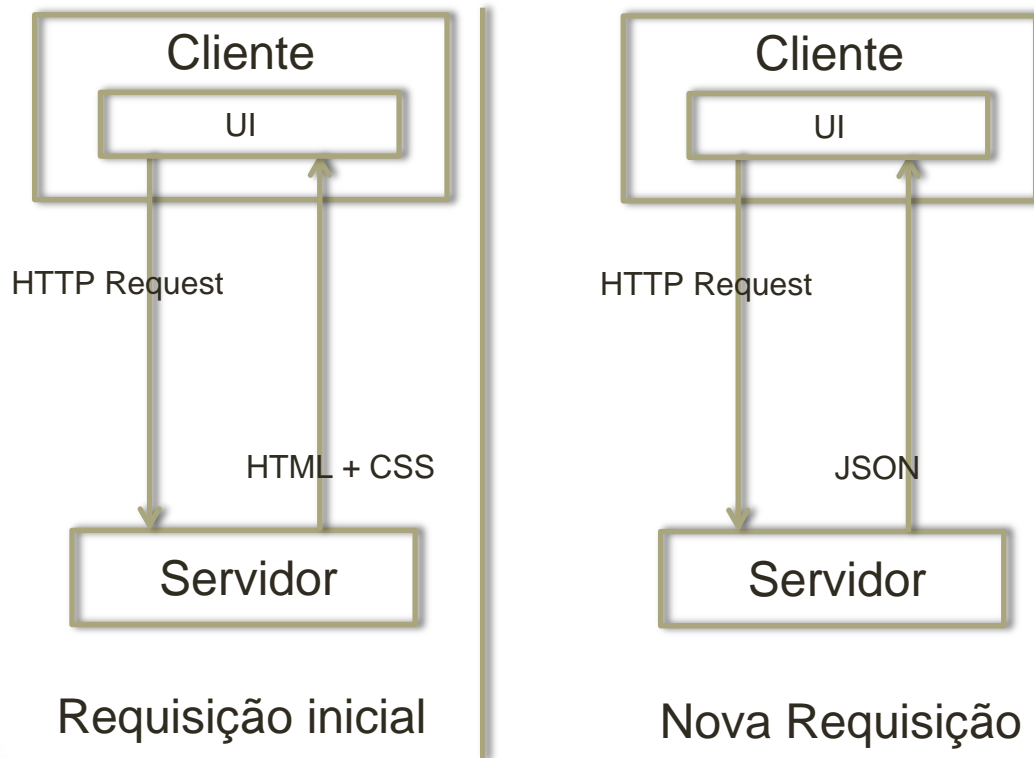
Dúvidas? Sugestões?
Comentários...

Vamos Praticar!!!

Angular

Angular

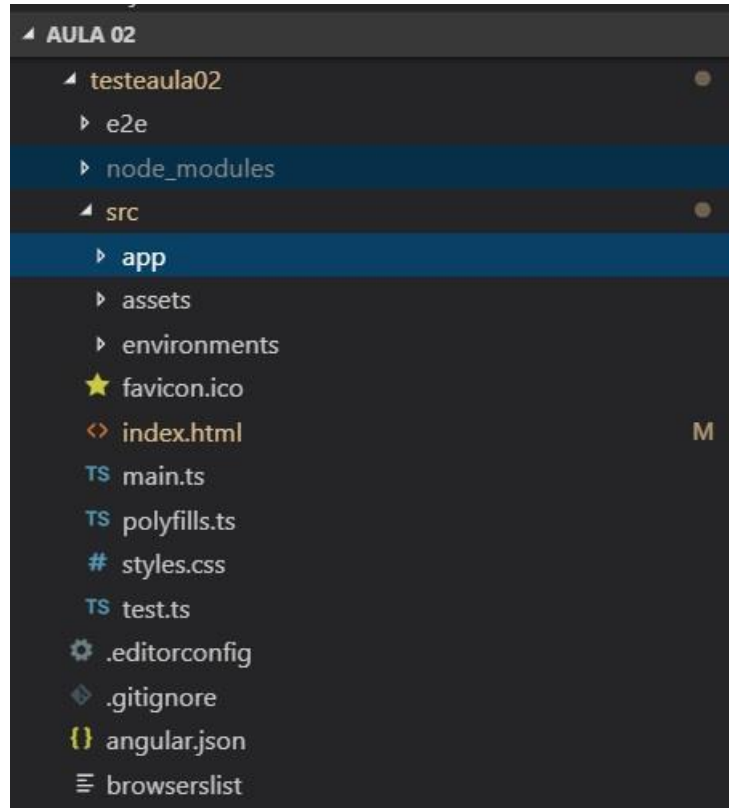
- Angular é um framework JavaScript open-source, criado pelo Google, que auxilia na execução de SPA;
- Versão 2.0 ou superior do AngularJS;
- Construído sob a ideologia de programação bem definida de separação do front (interface de usuário) e do back (regras de negócio);



Angular

- A arquitetura é composta de 4 partes principais:
 - Componentes (uma classe e um template responsáveis por uma parte da tela);
 - Diretivas (altera o DOM):
 - Estruturais (NgIf, ngFor);
 - Atributo (NgModel)
 - Serviços (encapsula a lógica do que não está envolvido com a parte visual)
 - Rotas (responsável pela navegação da aplicação)
- Angular.io (site oficial);
- Angular CLI (ferramenta de desenvolvimento) => <https://cli.angular.io/>;
- `npm install -g @angular/cli;`
- `ng new nome_projeto ;`
- `ng serve;`

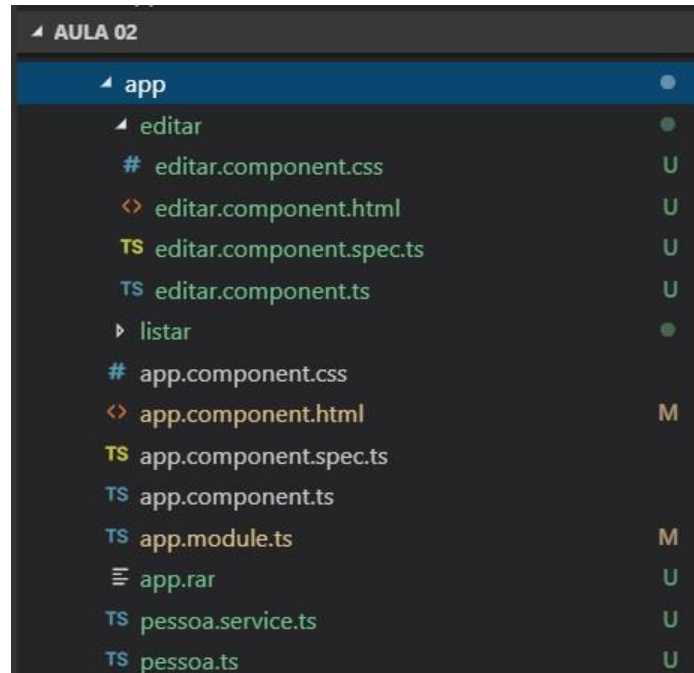
Angular



- `<app-root></app-root>` (index.html);

Angular

- Componentes definidos na pasta **app**



- app.component.ts (define a classe do componente);

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'testeaula02';
}
```

Angular

- Exemplo de template

```
<div>
  <button type="button" (click)="novo()">Novo</button>
  <table>
    <thead>
      <tr>
        <th>Nome</th>
        <th>Sobrenome</th>
        <th>Idade</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let pessoa of pessoas">
        <td><a href="javascript:;" (click)="editarPessoa(pessoa)">{{pessoa.nome}}</a></td>
        <td>{{pessoa.sobrenome}}</td>
        <td>{{pessoa.idade}}</td>
      </tr>
    </tbody>
  </table>
</div>
```

Angular

- Exemplo de classe

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Pessoa } from '../pessoa';
import { PessoaService } from '../pessoa.service';

@Component({
  selector: 'app-listar',
  templateUrl: './listar.component.html',
  styleUrls: ['./listar.component.css'],
  providers: [PessoaService]
})
export class ListarPessoasComponent implements OnInit {
  pessoas: Pessoa[];

  constructor(
    private router: Router,
    private service: PessoaService) { }

  ngOnInit() {
    this.pessoas = this.service.obterPessoas();
  }

  editarPessoa(pessoa: Pessoa) {
    this.router.navigate(['/pessoas', pessoa.id]);
  }

  novo() {
    this.router.navigate(['/pessoas/novo']);
  }
}
```

Angular

- Exemplo de template

```
<div>
  <div>
    <h5>Editar</h5>
  </div>
</div>
<div>
  <form method="get" (submit)="salvar(pessoa)">
    <div><label>Nome</label>
    <div>
      <input id="nome" name="id" [(ngModel)]="pessoa.nome" type="text">
    </div>
  </div>
  <div></div>
  <div><label>Sobrenome</label>
  <div>
    <input id="sobrenome" name="sobrenome" [(ngModel)]="pessoa.sobrenome" type="text">
  </div>
</div>
  <div></div>
  <div><label>Idade</label>
  <div>
    <input id="idade" name="idade" [(ngModel)]="pessoa.idade" type="text">
  </div>
</div>
  <button *ngIf="novo" type="submit">Criar</button>
  <button *ngIf="!novo" type="submit">Atualizar</button>
</form>
</div>
```

Angular

- Exemplo de classe

```
import { ActivatedRoute, Router } from '@angular/router';
import { PessoaService } from '../pessoa.service';
import { Pessoa } from '../pessoa';

@Component({
  selector: 'app-editar',
  templateUrl: './editar.component.html',
  styleUrls: ['./editar.component.css'],
  providers: [PessoaService]
})
export class EditarPessoaComponent implements OnInit {

  @Input()
  pessoa: Pessoa = new Pessoa();

  novo: boolean = false;

  constructor(
    private service: PessoaService,
    private route: ActivatedRoute,
    private router: Router) { }

  ngOnInit() {
    let id = +this.route.snapshot.params['id'];
    if(id) {
      this.pessoa = this.service.obterPessoa(id);
    } else {
      this.novo = true;
    }
  }

  salvar(pessoa: Pessoa) {
    this.router.navigate(['/pessoas']);
  }
}
```


Angular

- Exemplo de service

```
import { Injectable } from '@angular/core';
import { Pessoa } from './pessoa';

@Injectable()
export class PessoaService {
  private pessoas: Pessoa[] = new Array();

  constructor() { }

  obterPessoas(): Pessoa[] {
    let pessoa = new Pessoa();
    pessoa.id = 1;
    pessoa.nome = "Alex";
    pessoa.sobrenome = "Medeiros";
    pessoa.idade = 39;
    this.pessoas.push(pessoa);

    pessoa = new Pessoa();
    pessoa.id = 2;
    pessoa.nome = "Jose";
    pessoa.sobrenome = "Lima";
    pessoa.idade = 40;
    this.pessoas.push(pessoa);

    return this.pessoas;
  }

  obterPessoa(id: number): Pessoa {
    let pessoa = new Pessoa();
    pessoa.id = 1;
    pessoa.nome = "Alex";
    pessoa.sobrenome = "Medeiros";
    pessoa.idade = 39;
```

```
export class Pessoa {
  id: number;
  nome: string;
  sobrenome: string;
  idade: number;
}
```

Angular

- Rotas (app.module.ts)

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { RouterModule, Routes } from '@angular/router';

import { AppComponent } from './app.component';
import { ListarPessoasComponent } from './listar/listar.component';
import { EditarPessoaComponent } from './editar/editar.component';

const appRoutes: Routes = [
  { path: 'pessoas', component: ListarPessoasComponent },
  { path: 'pessoas/:id', component: EditarPessoaComponent }
];

@NgModule({
  declarations: [
    AppComponent,
    ListarPessoasComponent,
    EditarPessoaComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    RouterModule.forRoot(appRoutes)
  ],
  providers: [],
  bootstrap: [AppComponent]
})
```

- `<router-outlet></router-outlet>` e ``

Dúvidas? Sugestões?
Comentários...

Vamos Praticar!!!