

ANGULAR JS



OLHO NO CARRO
CONSULTA VEICULAR




Checktudo
Para a melhor decisão



Hello!

Ronaldo José Franciso Franco

✉ ronaldo.franco@olhonocarro.com.br

 [linkedin.com/ronaldo-j-f-franco](https://www.linkedin.com/ronaldo-j-f-franco)



OLHO NO CARRO
CONSULTA VEICULAR

Checktudo 
Para a melhor decisão



Conteúdo do curso

- ☐ Requisitos
- ☐ Curiosidades
- ☐ Componentes
- ☐ Roteamento
- ☐ Injeção de Dependências
- ☐ Diretivas
- ☐ O que é Angular?
- ☐ Utilização
- ☐ Metadata
- ☐ Data Binding
- ☐ Input Property
- ☐ Api Connection
- ☐ Blocos Principais
- ☐ Iniciando Aplicação
- ☐ Template
- ☐ Serviços
- ☐ Pipe

1

Requisitos



Conhecimento

HTML



JS

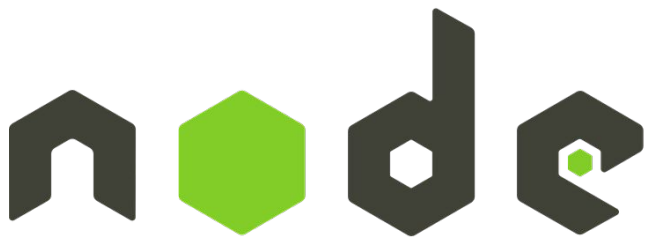


CSS





Dependências



nodejs.org



cli.angular.io

2

O que é Angular?



Origem

- É um framework JavaScript para o desenvolvimento de aplicações web criado e mantido pela Google
- Desenvolvido para facilitar o desenvolvimento de aplicações web SPA (Single Page Application)

3

Blocos Principais



Blocos Principais

Componentes

Metadata

Template

Roteamento

Data Binding

Serviços

Injeção de
dependência

Diretivas

4

Curiosidades



Curiosidades

- Não é continuação do Angular 1
- Open Source
- Parceria da Google com a Microsoft

5

Utilização



Forbes



6

Iniciando aplicação

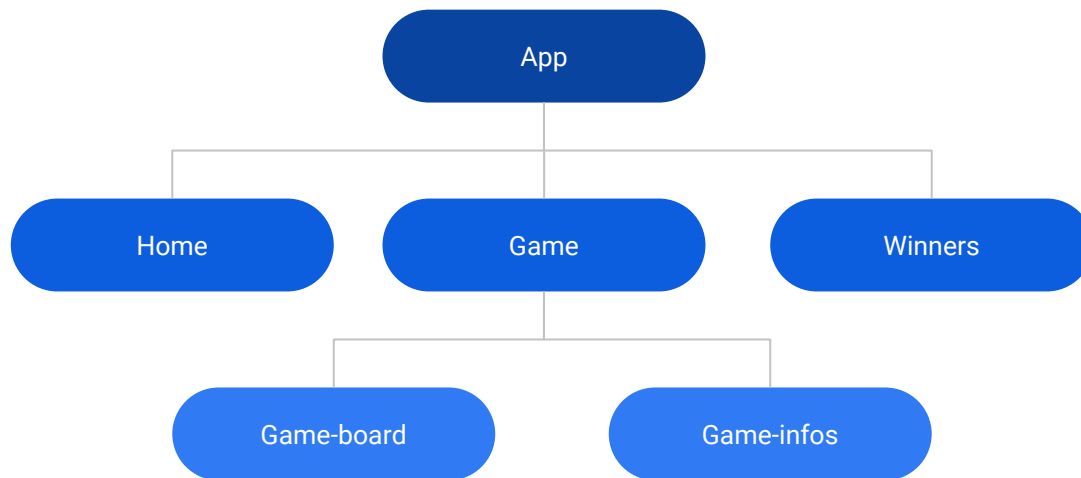


Nova aplicação

- `ng new "Nome da aplicação"`
- `ng serve.`



Estrutura



7

Componentes



Informações

- O Angular cria as interfaces por meio de componentes.
- Componentes nada mais são do que blocos de código.



Informações

- Componentes permitem a sua reutilização.
- Isso poupa tempo e traz praticidade no desenvolvimento
- `ng g c "nome do componente"`

8

Metadata



Informações

- São dados de configuração.
- Servem para definir utilização dos arquivos.
- Permitem o angular a ler e processar os arquivos

9

Template



Informações

- Corpo html do componente.
- Contém o esqueleto que a página irá renderizar

10

Roteamento



Informações

- Roteamento serve para realizar a transição entre componentes.
- Supre a falta de mais páginas na hora da navegação.

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { HomeComponent } from '../components/home/home.component';
import { GameComponent } from '../components/game/game.component';
import { WinnersComponent } from 'src/app/components/winners/winners.component';
```

```
const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'game', component: GameComponent },
  { path: 'winners', component: WinnersComponent },
];
```

```
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

11

Data Binding



Informações

- Permite a associação de informações do componente ao template.
- Existem 4 formas:
 - `{{ }}` interpolação
 - `[]` propriedade
 - `()` evento
 - `[()]` two way

```
7 <div class="container">
8   <div class="row">
9     <div class="col-md-9 col-lg-8 mx-auto">
10      <h3 class="login-heading mb-4">Jogo da Velha!</h3>
11      <div class="form-label-group">
12        <input type="text" id="name1" class="form-control" placeholder="Jogador 1" type="text"
13          [(ngModel)]="j1" required autofocus>
14        <label for="name1">Jogador 1</label>
15      </div>
16
17      <div class="form-label-group">
18        <input type="text" id="name2" class="form-control" placeholder="Jogador 2" type="text"
19          [(ngModel)]="j2" required autofocus>
20        <label for="name2">Jogador 2</label>
21      </div>
22
23      <a class="btn btn-lg btn-primary btn-block btn-login text-uppercase font-weight-bold mb-2"
24        routerLink="/game" [queryParams]="{player1:j1,player2:j2}" type="submit">jogar</a>
25
26      <a class="btn btn-lg btn-primary btn-block btn-login text-uppercase font-weight-bold mb-2"
27        routerLink="/winners" type="submit">Ganhadores</a>
28    </div>
29  </div>
30 </div>
```

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

```
@Component({  
  selector: 'app-home',  
  templateUrl: './home.component.html',  
  styleUrls: ['./home.component.css']  
})
```

```
export class HomeComponent implements OnInit {
```

```
  j1: string = "";
```

```
  j2: string = "";
```

```
  constructor() {}
```

```
  ngOnInit() {  
  }
```

```
}
```

12

Serviços



Informações

- São utilizadas para fazer a obtenção e transição de dados.
- Dados recebidos, ou dinâmicos devem ser tratados nas servisse, e depois providos.
- ng g s “nome da serviço”

13

Injeção de Dependência



Informações

- Basicamente faz ação de suprir necessidades do código.
- Injeta no código algo que realiza funções necessárias mas que não estão diretamente disponíveis

```
constructor(  
  private route: ActivatedRoute,  
  private _player: PlayerService,  
  private modalService: BsModalService,  
) {  
  this.player = _player;  
  this.inscricao = this.route.queryParams.subscribe(  
    (params: any) => {  
      this.j1 = { name: params['player1'], value: 1 };  
      this.j2 = { name: params['player2'], value: 2 };  
      this.player.setPlayer1(this.j1);  
      this.player.setPlayer2(this.j2);  
      this.player.setPlayerTurn(this.j1);  
    }  
  );  
}
```

14

Input Property

```
<div class="back">
  <game-infos [turn]="turn" [playerTurn]="playerTurn" [sec]="sec"></game-infos>

  <ng-template #template id="template">
    <div class="modal-header">
      <p class="modal-title pull-left"
        [ngClass]="(playerTurn == j1) ? 'j1w' : (playerTurn == j2) ? 'j2w' : 'velha'"
        {{finalMessage}}
      </p>
    </div>
    <div class="modal-body">
      Tempo de jogo {{sec | timer}}
    </div>
    <div class="modal-footer pull-left">
      <button type="button" class="btn btn-info pull-right" (click)="reload();modalRef.hide()">
        Novo Jogo
      </button>
      <button type="button" class="btn btn-danger pull-right" (click)="modalRef.hide()" routerLink="/">
        Inicio
      </button>
    </div>
  </ng-template>
</div>
```

```
import { Component, OnInit, Input } from '@angular/core';
import { Player } from '../model/player';
```

```
@Component({
  selector: 'game-infos',
  templateUrl: './game-infos.component.html',
  styleUrls: ['./game-infos.component.css']
})
export class GameInfosComponent implements OnInit {

  @Input()
  turn: number;

  @Input()
  playerTurn: Player;

  @Input()
  sec: number;

  constructor() {

  }

  ngOnInit() {}

}
```

15

Pipe



Informações

- Forma de manipular a apresentação de dados.
- `ng g p "nome do pipe"`


```
import { Pipe, PipeTransform } from '@angular/core';
```

```
@Pipe({  
  name: 'timer'  
})
```

```
export class TimerPipe implements PipeTransform {
```

```
  transform(value: number): string {
```

```
    const minutes: number = Math.floor(value / 60);
```

```
    return (minutes>0) ? minutes + ' min ' + (value - minutes * 60) + ' seg' : value + ' seg';
```

```
  }
```

```
}
```

16

Diretivas



Informações

- São atributos que alteram o comportamento dos elementos HTML.
- Existem dois tipos:
 - Diretivas estruturais
 - Diretivas de atributos
- ng g d “nome da diretiva”

```
import { Directive, EventEmitter, ElementRef, HostListener, Renderer2, Input, Output } from '@angular/core';
import { PlayerService } from '../services/player.service';
import { Player } from '../model/player';

@Directive({
  selector: '[SetPoint]'
})

export class SetPointDirective {

  @Input()
  SetPoint: Player;
  image: string;

  @Output() point = new EventEmitter();
  constructor(private el?: ElementRef, private renderer?: Renderer2, private _player?: PlayerService) {
  }

  @HostListener('click')
  onClick() {
    if (this.el.nativeElement.getAttribute('value')) event.stopPropagation();
    else {
      this.image = (this.SetPoint.value == 1) ? '../assets/img/x.png' : '../assets/img/o.png';
      this.renderer.setStyle(this.el.nativeElement, 'background-image', `url('${this.image}')`);
      this.renderer.setStyle(this.el.nativeElement, 'background-size', "contain");
      this.renderer.setStyle(this.el.nativeElement, 'background-position', "center");
      this.renderer.setStyle(this.el.nativeElement, 'background-repeat', "no-repeat");
      this.renderer.setAttribute(this.el.nativeElement, 'value', `${this.SetPoint.value}`);
      this._player.changePlayer();
      this.point.emit({campo : this.el.nativeElement.getAttribute('name')});
    }
  }
}
```

```
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import { Player } from '../../model/player';

@Component({
  selector: 'game-board',
  templateUrl: './game-board.component.html',
  styleUrls: ['./game-board.component.css']
})
export class GameBoardComponent implements OnInit {

  constructor() {

  }

  ngOnInit() {

  }

  @Input() playerBoard: Player;
  @Output() point = new EventEmitter();

  change(event){
    this.playerBoard = event;
    this.point.emit(this.playerBoard);
  }
}
```

```
<div class="back">
  <game-infos [turn]="turn" [playerTurn]="playerTurn" [sec]="sec"></game-infos>
  <game-board [playerBoard]="playerTurn" (point)="attPlayer($event)"></game-board>

  <ng-template #template id="template">
    <div class="modal-header">
      <p class="modal-title pull-left"
        [ngClass]="(playerTurn == j1) ? 'j1w' : (playerTurn == j2) ? 'j2w' : 'velha'"
        {{finalMessage}}
      </p>
    </div>
    <div class="modal-body">
      Tempo de jogo {{sec | timer}}
    </div>
    <div class="modal-footer pull-left">
      <button type="button" class="btn btn-info pull-right" (click)="reload();modalRef.hide()">
        Novo Jogo
      </button>
      <button type="button" class="btn btn-danger pull-right" (click)="modalRef.hide()" routerLink="/">
        Inicio
      </button>
    </div>
  </ng-template>
</div>
```

17

API Connection

```
import { HttpClient } from '@angular/common/http';
import { Player } from 'src/app/model/player';
import { Winner } from 'src/app/model/winner';
import { tap, take } from 'rxjs/operators';

@Injectable({
  providedIn: 'root'
})
export class WinnersService {

  private readonly API = 'https://vast-spire-73802.herokuapp.com/winners';

  constructor(private http: HttpClient) { }

  getWinners(){
    return this.http.get<Winner[]>(this.API)
      .pipe(
        tap(console.log)
      );
  }

  addWinner(player: Player, time: number){

    return this.http.post(this.API,{name: player.name, time: time})
      .pipe(take(1));
  }
}
```



```
<h1>Vencedores!!</h1>
<div *ngIf="winners.length > 0 else vazio" class="col-12 row container">
  <div class="col-6">
    <ul class="list-group">
      <li *ngFor="let winner of winners" class="list-group-item ">
        <div class="col-6">
          <span class="title">Jogador:</span> {{winner.name}}
        </div>
        <div class="col-6">
          <span class="title">Tempo:</span> {{winner.time | timer}}
        </div>
      </li>
    </ul>
  </div>
  <div class="col-6 bg-image"></div>
</div>

<ng-template #vazio>
  <h4>Ainda não há vencedores</h4>
</ng-template>
```

```
import { Component, OnInit } from '@angular/core';
import { WinnersService } from 'src/app/services/winners.service'
import { Winner } from 'src/app/model/winner';
```

```
@Component({
  selector: 'app-winners',
  templateUrl: './winners.component.html',
  styleUrls: ['./winners.component.css']
})
export class WinnersComponent implements OnInit {

  winners: Winner[];

  constructor(private _winners: WinnersService) {
    const request = this._winners.getWinners()
      .subscribe(dados =>
        this.winners = dados.sort((n1, n2) => {
          if (n1.time > n2.time) return 1;
          else if (n1.time < n2.time) return -1;
          return 0;
        })
      );
    setTimeout(() => request.unsubscribe(), 2000);
  }
  ngOnInit() { }
}
```

```
attPlayer(event) {  
  this.setBoard(event);  
  if (this.turn > 4) this.finish = this.checkBoard();  
  if (this.finish){  
    this.finalMessage = `${this.playerTurn.name} Venceu !!`;   
    clearInterval(this.interval);  
  
    this.winners.addWinner(this.playerTurn, this.sec).subscribe();  
  
    this.openModal(this.myModal);  
  }  
  else if(this.turn == 9){  
    this.finalMessage = 'Deu Velha !!';  
    clearInterval(this.interval);  
    this.openModal(this.myModal);  
    this.playerTurn = new Player;  
  }  
  else{  
    this.turn++;  
    this.playerTurn = this.player.getPlayer();  
  }  
}
```



Obrigado!

✉ ronaldo.franco@olhonocarro.com.br

in [linkedin.com/ronaldo-j-f-franco](https://www.linkedin.com/ronaldo-j-f-franco)

🐙 github.com/RonaldoJFFranco/cursoFinal

🐙 github.com/RonaldoJFFranco/curso



OLHO NO CARRO
CONSULTA VEICULAR

Checktudo ✓
Para a melhor decisão