

DESAFIO 9



Com base no curso de “Angular 11 Avançado: Criando uma Arquitetura Master/Detail” responda as seguintes questões.

- a) Qual a responsabilidade do package.json no projeto?
- b) Qual a responsabilidade do angular.json no projeto?
- c) Qual a finalidade da tag <router-outlet></router-outlet>?
- d) Qual a diferença entre as duas declarações de rotas a seguir? Qual delas é denominada de eager-load e lazy-load?

Declaração 1:

```
const routes: Routes = [  
  { path: 'entries/new', component: EntryFormComponent },  
];
```

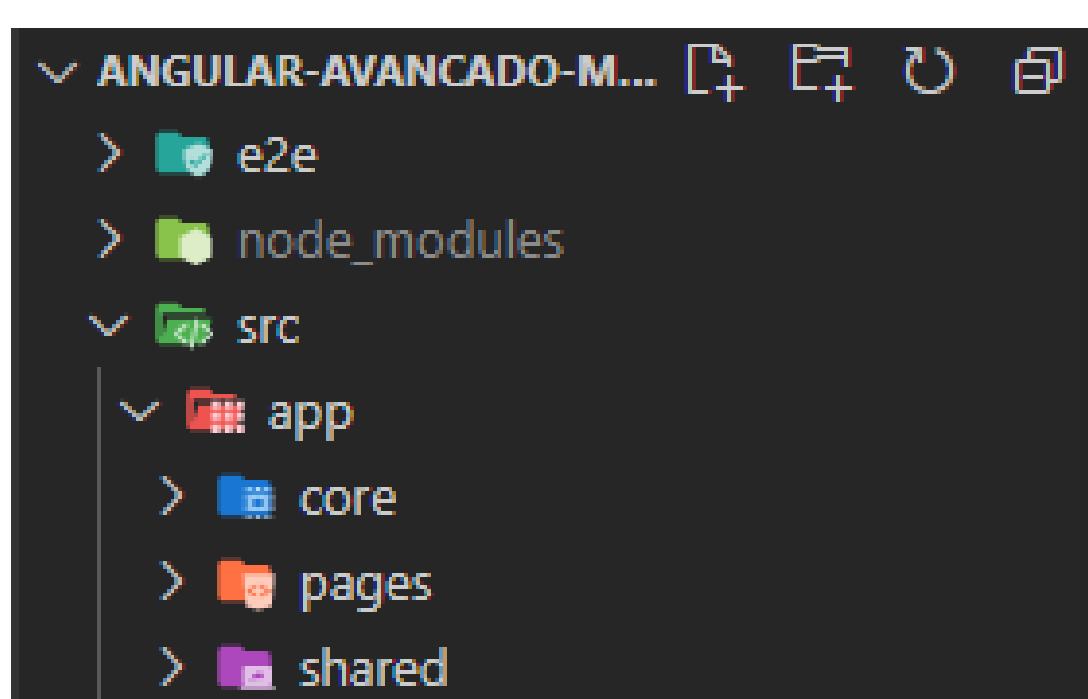
Declaração 2:

```
const routes: Routes = [  
  { path: 'entries', loadChildren: () => import('./pages/entries/entries.  
module').then(m => m.EntriesModule) },  
];
```

- e) Qual a importância da componentização?
- f) Quais componentes reaproveitáveis foram criados no projeto?
- g) No arquivo bread-crumbs.component.ts qual funcionalidade da linha a seguir:

```
@Input() items: Array<BreadCrumbItem> = [];
```
- h) Com base no exercício anterior para qual finalidade a anotação a seguir é utilizada?

```
@Output()
```
- i) Sobre a arquitetura proposta no curso, descreva os requisitos para se enquadrar em cada um dos níveis a seguir: **core**, **pages** e **shared**



- j) Qual o papel do arquivo in-memory-database.ts?
- k) Liste 5 métodos e 5 propriedades de um objeto do tipo FormBuilder e suas respectivas funcionalidades. Segue exemplo abaixo de declaração.

```
this.categoriaForm = this.formBuilder.group({  
  id: [null],  
  name: [null, [Validators.required, Validators.minLength(2)]],  
  description: [null]  
});
```

l) Explique o funcionamento da seguinte sentença:

```
if (this.currentAction == "edit") {  
  this.route.paramMap.pipe(  
    switchMap(params => this.lancamentoService.getById(+params.get("id")))  
  )  
  .subscribe(  
    (lancamento) => {  
      this.lancamento = lancamento;  
      this.lancamentoForm.patchValue(lancamento);  
    },  
    (error) => alert('Ocorreu um erro no servidor, tente mais tarde.')  
  )  
}
```

m) A linha a seguir se refere a encapsulamento, polimorfismo, herança, abstração ou injeção de dependências?

```
export class EntryFormComponent extends BaseResourceFormComponent<Entry>  
implements OnInit
```

n) A linha a seguir se refere a encapsulamento, polimorfismo, herança, abstração ou injeção de dependências?

```
ngOnInit() {  
  this.loadCategories();  
  super.ngOnInit();  
}
```

o) A linha a seguir se refere a encapsulamento, polimorfismo, herança, abstração ou injeção de dependências? Existe outra maneira?

```
@NgModule({  
  providers: [  
    EntryService  
  ]  
})
```

01

GIT

02

HTML+CSS

03

FLEX-GRID

04

PROTÓTIPO

5.1

SPA

5.2

COMPONENTS

5.3

SERVICES

06

ANGULAR

07

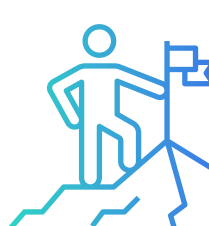
DESIGN SYSTEM

08

INTEGRAÇÃO

09

DESAFIO FINAL



ANDAMENTO DO CURSO
ANGULAR 11 AVANÇADO:

SEÇÃO 3 até 4