

Angular



- **Framework** baseado em Javascript
- **Typescript** como linguagem principal
- **SPAs** são o seu foco
- **Componentes** são a base da arquitetura
- **Estrutura organizada** e bem definida
- **Sistema de Tooling** muito rico: CLI, HTTP, Router..

PRÉ REQUISITOS



- **HTML & CSS**
- **NODE & NPM**
- **JAVASCRIPT & TYPESCRIPT**

ANGULARJS VS ANGULAR



ANGULARJS
by Google

!=



Initial release date: October 20, 2010

Developer(s): Google

Written in: JavaScript

Developer(s): Google

Initial release: 2.0 / 14 September 2016; 5 years ago

License: MIT License

Stable release: 14.1.0 / 20 July 2022; 17 days ago

Written in: TypeScript

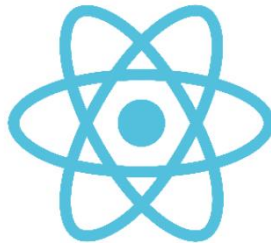
</> MyApp Home View 1 View 2

Hello, Angular

 [Learn about this Angular VS.NET template](#)

</> React App! Home View 1 View 2

Hello, React!

 [Learn about this React VS.NET template](#)

</> Aurelia App! Home View 1 View 2

Hello, Aurelia!

 [Learn about this Aurelia VS.NET template](#)

</> Vue App! Home View 1 View 2

Hello, Vue!

 [Learn about this Vue VS.NET template](#)



VS



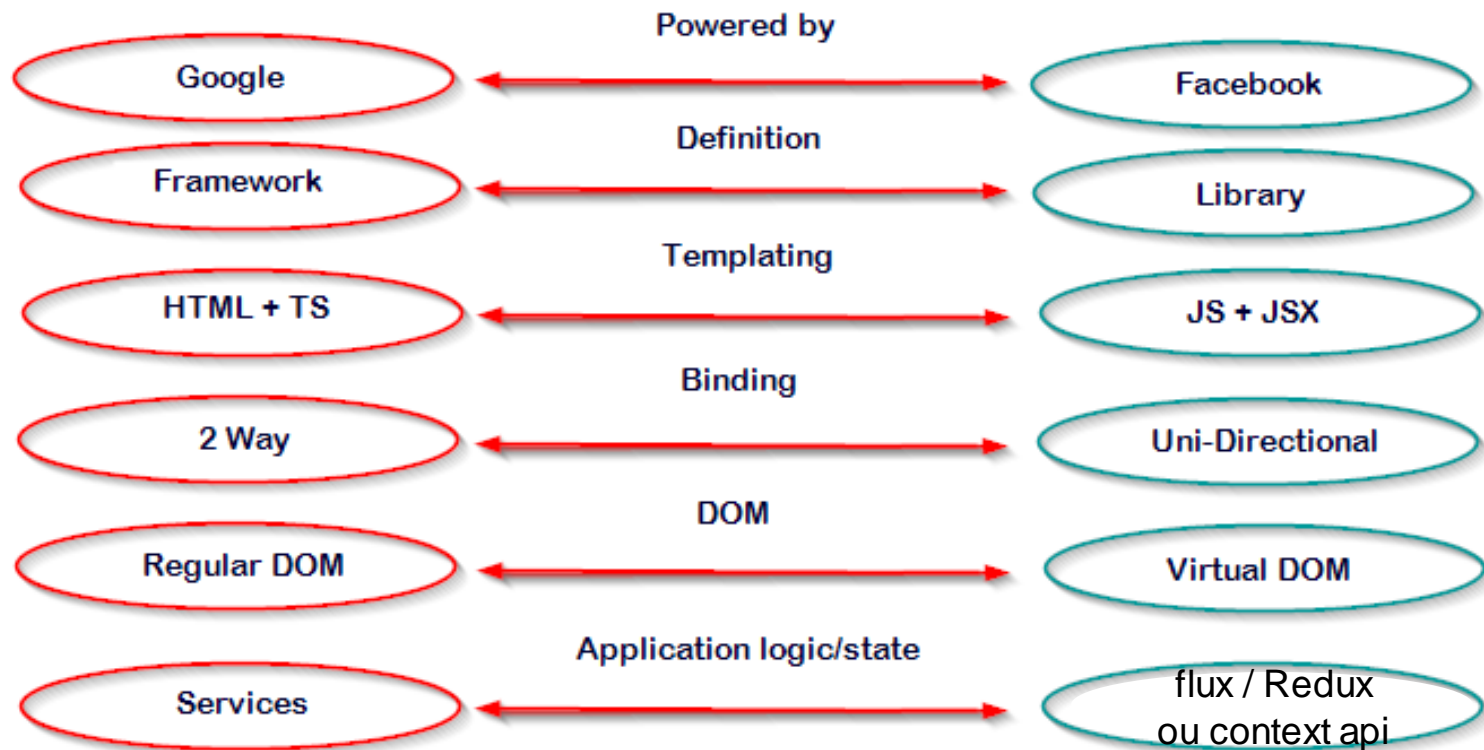
React



Angular



VS



WEB
APP

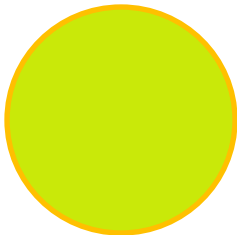
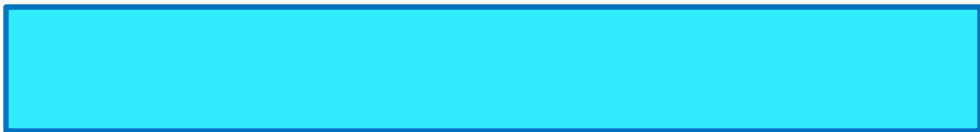
COMPONENTES

GERENCIAMENTO DE
ESTADO

ROTEAMENTO

RENDERIZAÇÃO

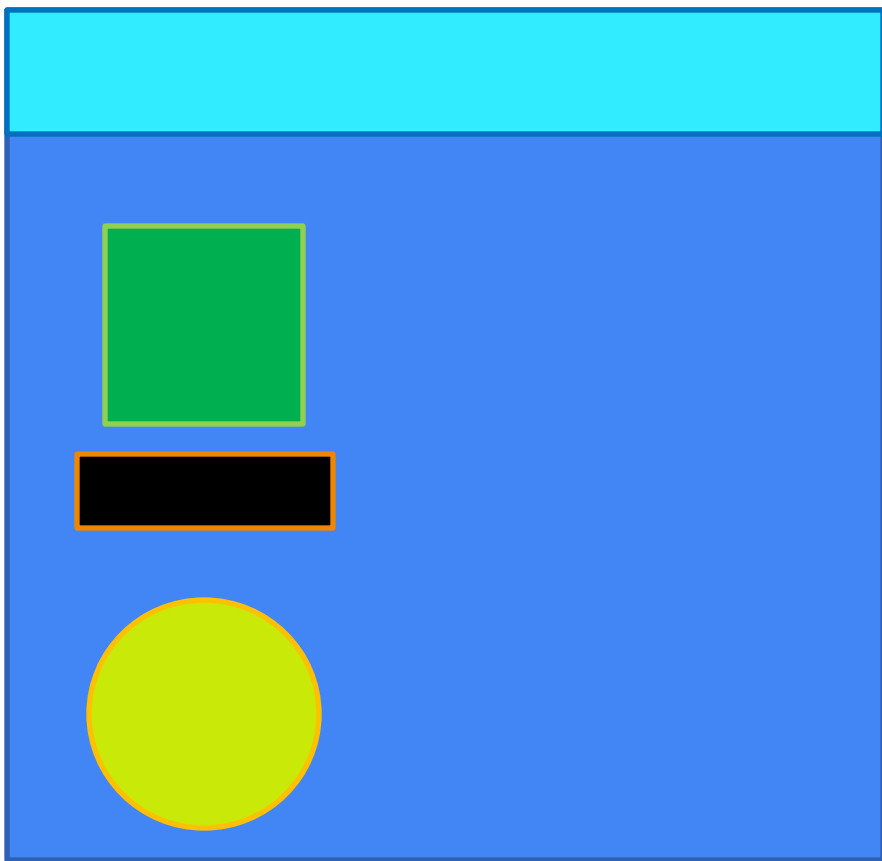
Componentes



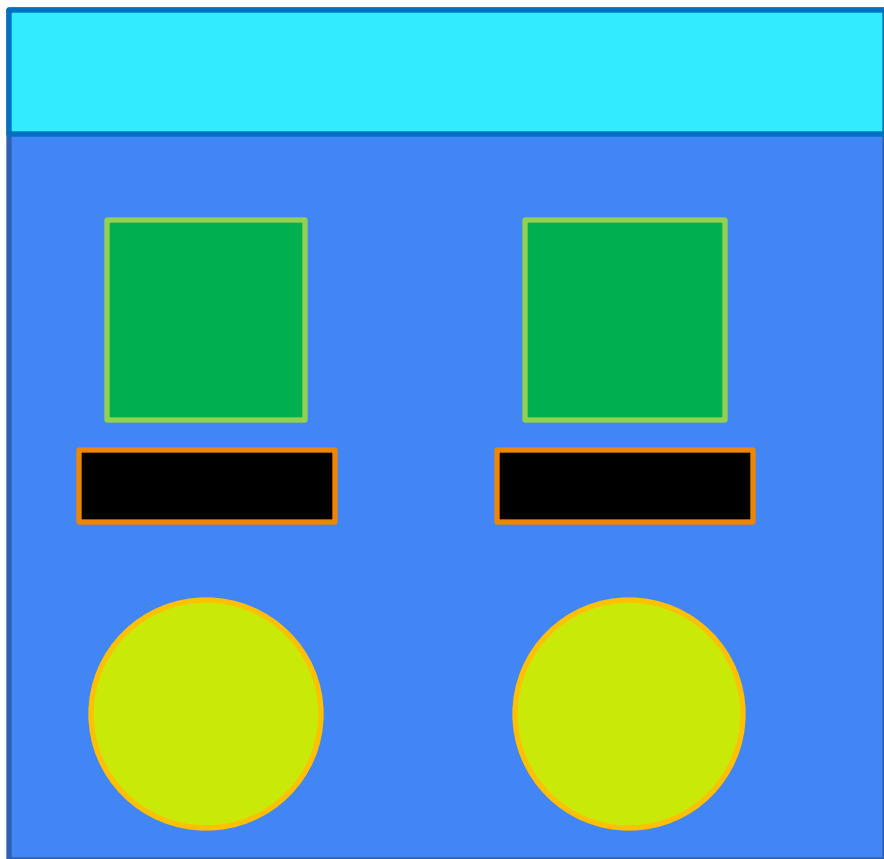
Componente é um elemento **visual**, **customizável** e **reutilizável**

Exemplo: um card na tela, botão

Componentes



Componentes



Gerenciamento de estados

states

Responsável por garantir que as informações dos componentes não estão dessincronizadas e facilitar a comunicação de um componente com outro

Exemplo: Atualizar carrinho de compras ao clicar no botão de comprar de um produto



Comprar agora

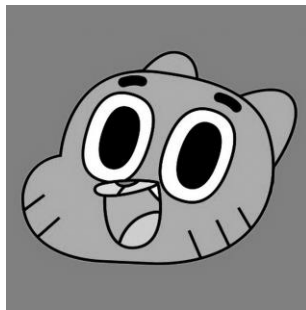
Adicionar ao carrinho

Gerenciamento de estados

states



disponível



indisponível

Se eu estiver cinza,
Deixe o outro
componente cinza

```
{  
  Cor: verde  
  Texto: "disponível"  
}
```

```
{  
  Cor: cinza  
  Texto: "indisponível"  
}
```

Gerenciamento de estados



states

EXEMPLO DE FERRAMENTAS PARA GERENCIAMENTO DE ESTADO:

- FLUX
- REDUX
- CONTEXT API
- RECOIL
- STOREON
- VUEX
- NGRX
- NGXS

Roteamento

Routes ou Routing & Navigation

Responsável pela forma de navegar fazendo a troca de URL sem Recarregar a página, mudando somente os componentes que são mostrados

meuapp/login

meuapp/home

Roteamento

Routes ou Routing & Navigation



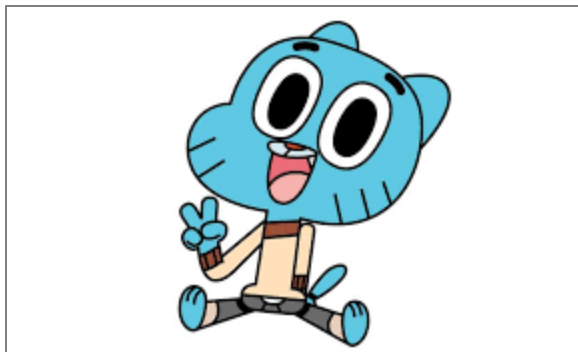
meuapp/login



EMAIL

SENHA

meuapp/home



“bem vindo Gunball”

Roteamento

Routes ou Routing & Navigation

Responsável pela forma de navegar fazendo a troca de URL sem Recarregar a página, mudando somente os componentes que são mostrados

Exemplos de bibliotecas:

- History API
- React-Router
- Vue-Router
- Angular RoutingModule

Renderização

Render

Responsável por decidir a melhor maneira de acessar e entregar para o browser o componente pronto para que ele seja desenhado na janela sem complicações



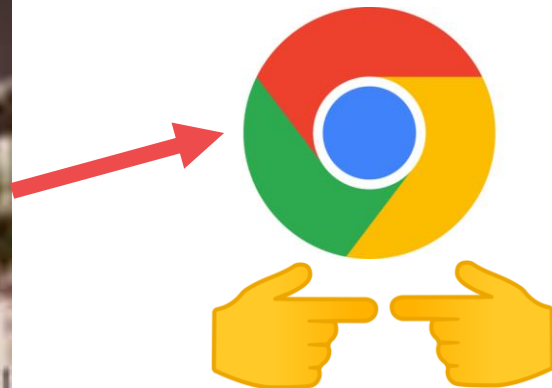
ROUTES

COMPONENTS

STATES



Muito obrigado

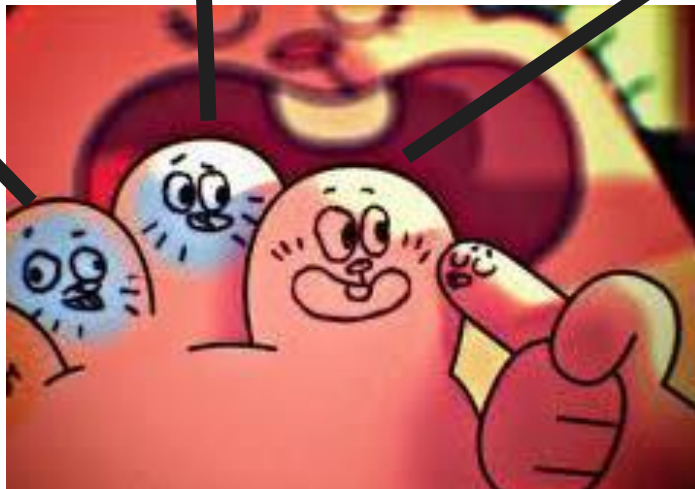


Renderização

Render

Existem 3 tipos principais estratégias de renderização

- 100% Server
- Parte server e Parte Client
- 100% Client



COMPONENTES



GERENCIAMENTO DE ESTADO



ROTEAMENTO



RENDERIZAÇÃO



SEU CÓDIGO

MODELO DE COMPONENTE

TEMPLATES

EM QUAL PASTA COLOCAR

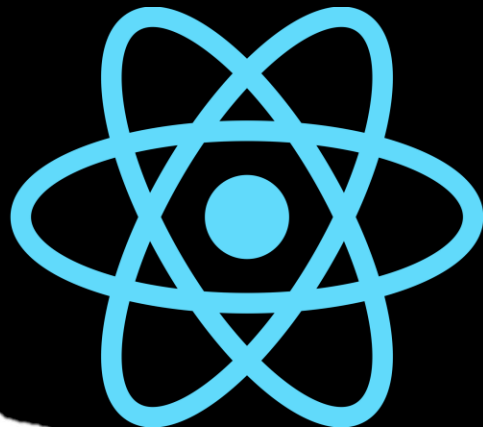
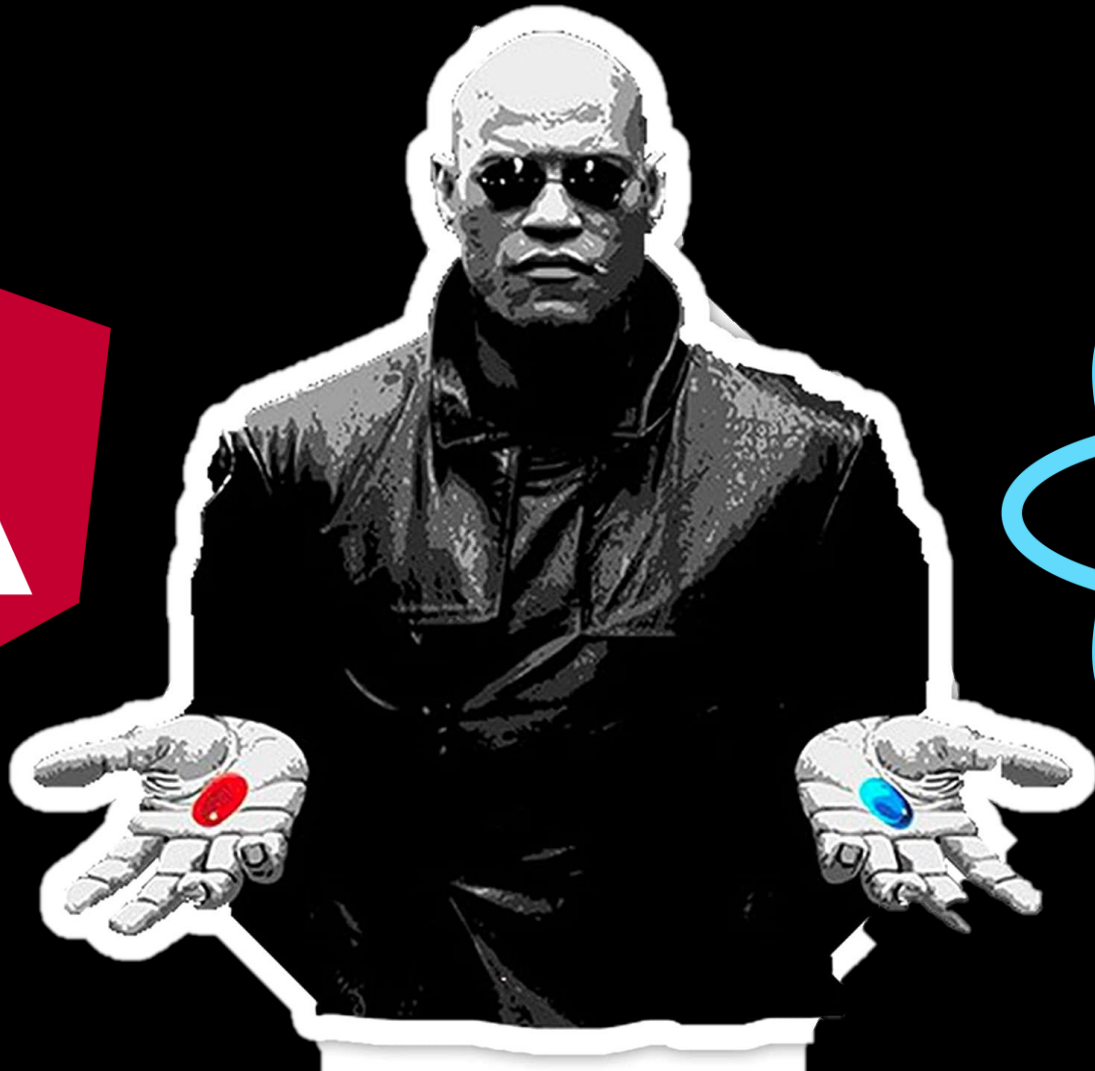
ARQUIVOS GERAIS

NG



SEU CÓDIGO

```
dracula.js  x  settings.json
home > difford > repos > _orgs > dracula > dracula-vscode > dracula.js > ...
1  /*
2  * Once upon a time ...
3  */
4
5  class Vampire {
6    constructor(props) {
7      this.location = props.location;
8      this.birthDate = props.birthDate;
9      this.deathDate = props.deathDate;
10     this.weaknesses = props.weaknesses;
11   }
12
13   get age() {
14     return this.calcAge();
15   }
16
17   calcAge() {
18     return this.deathDate - this.birthDate;
19   }
20 }
21
22 // ... there was a guy named Vlad
23
24 const Dracula = new Vampire({
25   location: 'Transylvania',
26   birthDate: 1428,
27   deathDate: 1476,
28   weaknesses: ['Sunlight', 'Garlic']
29 });
```

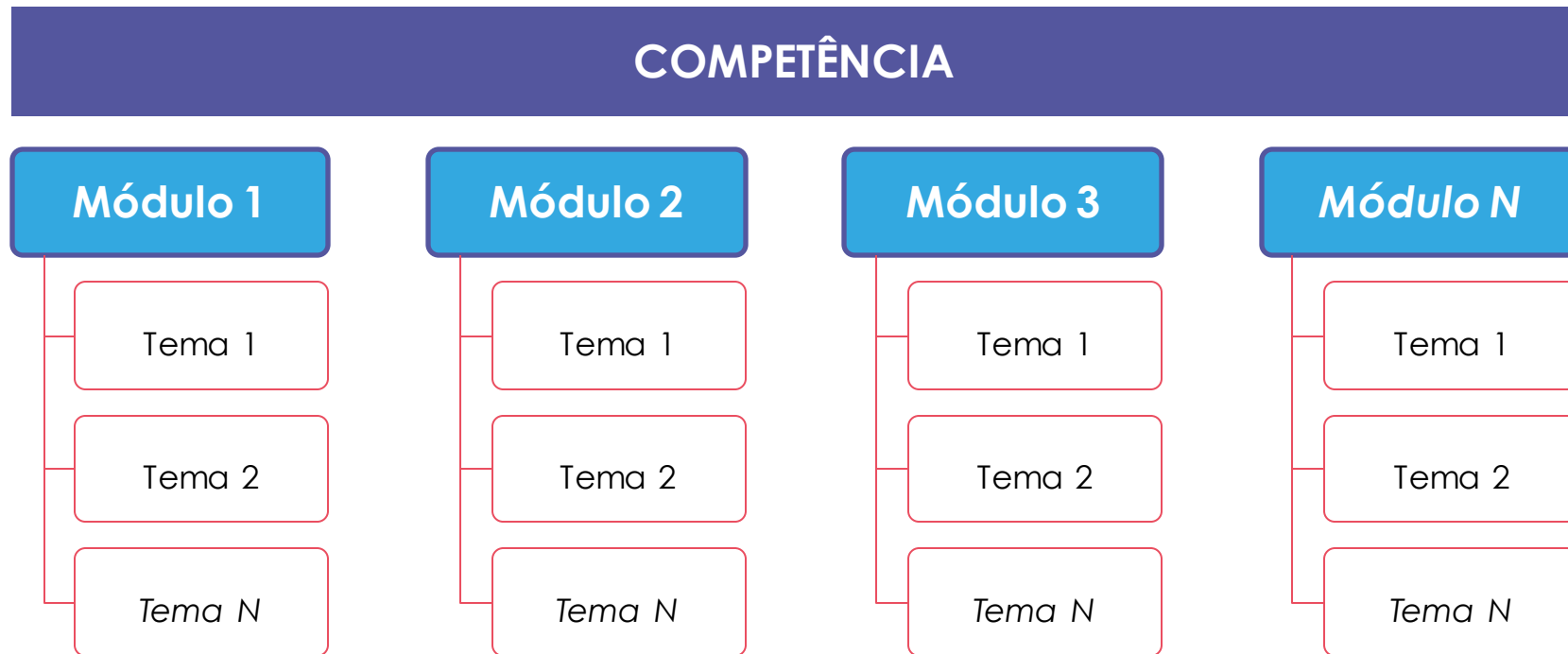


Espaço da webcam

Em todos os slides, evite escrever ou usar imagens que possam ocupar a área mostrada ao lado, pois ela representa o espaço reservado para a webcam.

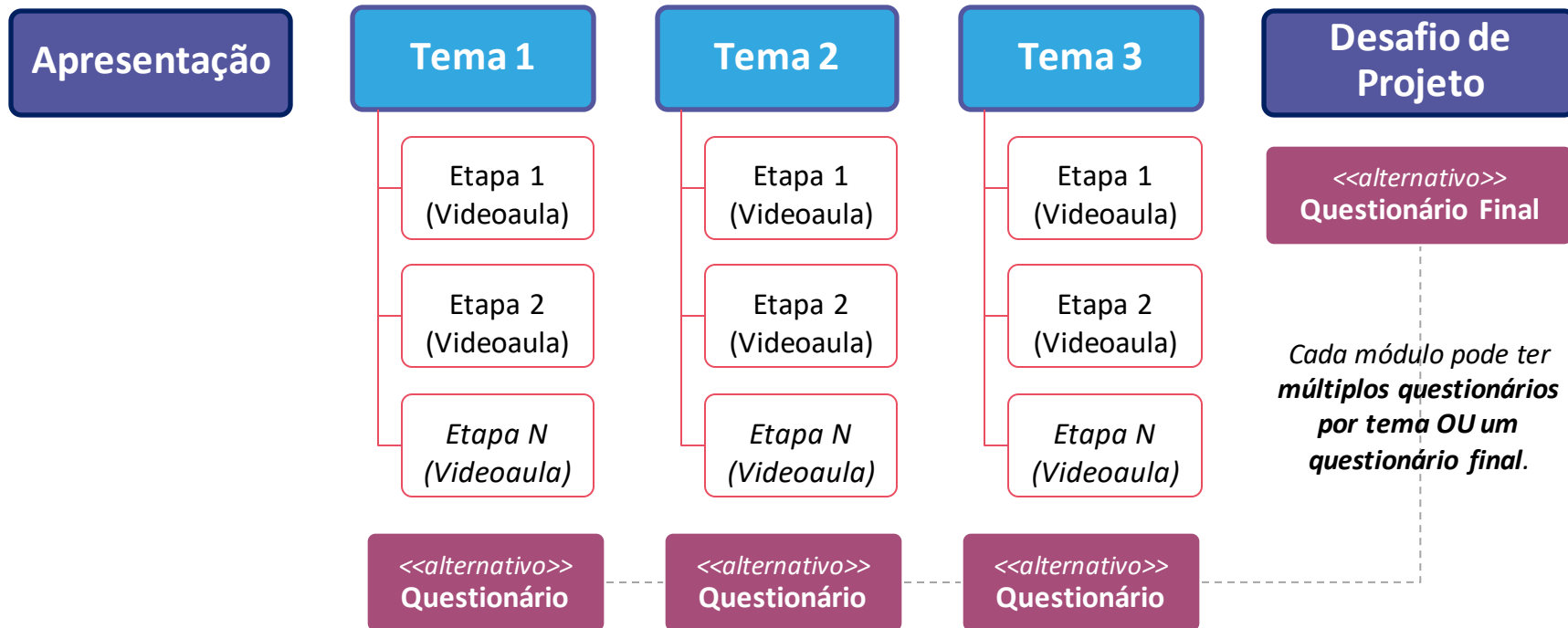
WEBCAM

Estrutura das Trilhas



Nesse contexto, "**Módulos**" são "**Subcompetências**".

Estrutura dos Módulos



Cada **Etapa** deve ter, **idealmente, em torno de 15 minutos** (isso pode variar, principalmente em videoaulas práticas). Analogamente, recomendamos que cada **Tema** tenha **entre 2 e 4 horas**.

Escreva o Título do Tema

Cada tema deve ter sua respectiva apresentação, visando o desenvolvimento (teórico e/ou prático) do tópico em questão.

Seu nome aqui

Sua profissão e informações profissionais

@redes

Sobre Mim

- Conte um pouco da sua experiência
- Como entrou na área
- O que te motiva
- O que você faz além de programar (hobby, etc)
- Como te encontram: linkedin, GitHub, etc..)

Se preferir, pode utilizar elementos visuais para contar um pouco da sua jornada, como uma linha do tempo, imagens, fotos e etc.

Objetivo Geral

Escreva aqui qual é o objetivo geral do tema; o que o Dev será capaz de fazer ao final e sobre a importância do conteúdo.

Pré-requisitos

- Liste aqui os pré-requisitos para o tema, desde configurações do ambiente até as noções básicas necessárias para uma melhor assimilação do conteúdo

Percurso

Etapa 1

Título da Etapa 1

Etapa 2

Título da Etapa 2

Etapa 3

Título da Etapa 3

Módulo 1

Fundamentos do desenvolvimento Web

Otávio Perkles - @operkles

Tópico da Etapa 1

Conteúdo...

Percurso

~~Etapa 1~~

~~Título da Etapa 1~~

Etapa 2

Título da Etapa 2

Etapa 3

Título da Etapa 3

Etapa 2

Título da Etapa 2

// Título do Tema/Apresentação

Tópico da Etapa 2

Conteúdo...

Percurso

~~Etapa 1~~

~~Título da Etapa 1~~

~~Etapa 2~~

~~Título da Etapa 2~~

Etapa 3

Título da Etapa 3

Etapa 3

Título da Etapa [Exemplo Prático]

// Título do Tema/Apresentação

Hands On! Exemplo de Slide de Transição (Prática)

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Links Úteis

- Repositório no GitHub
- Documentação Oficial
- Referências:
 - Link X
 - Link Y

Adicione aqui links úteis como o Repositório no GitHub, Documentação Oficial, Referências e materiais complementares.

Para saber mais

Caso você ainda sinta que possa deixar informações úteis para os devs, como um site na Web, um curso da Dio, livros ou temas para se aprofundar no assunto. Coloque-os aqui.

Lembre-se de que você é uma referência para àqueles que vão te assistir.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)

