2ª Aula – LPW – Glauco

Baixar node;

**Estrutura da WEB**

A programação web é baseada em 3 braços HTML, CSS e Javascript.

HTML é a estrutura

CSS é o estilo, organização

Javascript é o comportamento

O Navegador é quem executa a programação web, navegadores exemplos como Google, Safari, Edge,

Qual é o motor do Chrome? V8

O que é o Chrome V8?

O Chrome V8 executa código JavaScript. O Node.js é baseado no Chrome V8 e é um ambiente de tempo de execução amplamente usado para funções JavaScript sem servidor. No entanto, há vantagens em executar funções diretamente no V8.

JavaScript eu faço aplicações no front-end, o navegador apenas entende JavaScript mas se utiliza de frameworks como Angular, React.js, Vue.js, para ser mais produtivo.

Node.js é o Javascript que roda no lado do servidor e pode ser utilizado outras linguagens como JAVA, Python utilizando ecossistema Spring e Django respectivamente.

**O que é um framework?**

Um framework é um **conjunto de ferramentas e bibliotecas** que fornecem um conjunto de **recursos e funcionalidades** prontas para desenvolvedores e desenvolvedoras de software.

Isso permite construir aplicativos mais rapidamente, com menos esforço e código, e com menores chances de erros.

**Angular**

Angular é um framework, Ele é um alicerce já montado que eu organizo meu código em cima dessa estrutura. Tudo que eu enxergo na tela usando Angular vira um componente. Um componente é formado por Html, CSS e TypeScript que é uma classe, Angular é um framework orientado a objetos. StoryBook é um organizado de componente.

Um componente Angular é o principal bloco de construção de uma aplicação Angular. Ele encapsula uma parte da interface do usuário e sua lógica associada, promovendo a reutilização de código e a organização da aplicação.

**Um componente Angular é composto por três partes principais:**

* **Template:** A parte visual do componente, escrita em HTML. É aqui que você define a estrutura da interface do usuário, como botões, inputs, listas, etc.
* **Style:** A parte de estilização do componente, escrita em CSS. Aqui você define as regras de estilo que serão aplicadas aos elementos do template.
* **Class:** A parte lógica do componente, escrita em TypeScript. É aqui que você define as propriedades, métodos e comportamentos do componente.

**Para que serve cada parte?**

* **Template:** Define a estrutura visual do componente.
* **Style:** Define a aparência do componente.
* **Class:** Controla a lógica do componente, como:
  + **Propriedades:** Armazenam os dados do componente.
  + **Métodos:** Realizam as ações do componente, como manipular eventos, fazer chamadas a serviços, etc.
  + **Inputs:** Recebem dados de outros componentes.
  + **Outputs:** Emitem eventos para outros componentes.

**Benefícios de utilizar componentes:**

* **Reutilização de código:** Crie componentes uma vez e utilize-os em diversos locais da aplicação.
* **Organização do código:** Divida a aplicação em partes menores e mais fáceis de entender e manter.
* **Teste:** Facilita a criação de testes unitários para cada componente.
* **Manutenção:** Permite fazer alterações em um componente sem afetar outros componentes.

**Em resumo:**

Um componente Angular é uma unidade de código que encapsula uma parte da interface do usuário e sua lógica. Ao entender como os componentes são construídos e como suas partes se relacionam, você estará mais preparado para criar aplicações Angular robustas e escaláveis.

**React.js**

React.js é uma coletânea de bibliotecas, que precisam de mais de uma biblioteca para fazer uma operação. Eu construo meu alicerce a partir de bibliotecas.

React é um [**framework JavaScript**](https://www.alura.com.br/artigos/angular-vs-react-vs-vue-js) criado pelo Facebook (atual **Meta**) que é usado para criar [interfaces de usuário (UI)](https://www.alura.com.br/artigos/ux-e-ui-conheca-as-semelhancas-e-diferencas-entre-ambos) em aplicativos web.

Ele é **popular** por ser fácil de usar, altamente **flexível** e **escalável**, e é usado por muitas empresas de tecnologia, incluindo o Facebook, Instagram e Airbnb.

Single Page Aplication

É uma maneira de visualizar o site, carregando HTML, CSS e JavaScript e apenas carrega dados e pede dados.

Exemplo: Tenho uma página que mostra uma lista de produtos os meus componentes já estão preparados ele vai apenas puxar dados no back-end utilizando JSON ou XML que vai apenas retornar dados.

Pagamentos {

Id:

Descrição:

Preço:

}

Quando o componente solicita os dados para poder mostrar em sua página web, esses dados são em formato JSON ou XML, o Angular vai fazer uma requisição o Request HTTP para poder pegar esses dados no Back-End e volta um Response HTTP.

Por detrás dos sites, como se transferem dados?

HTTP 🡪 É um HyperText Transfer Protocol;

TCP 🡪 É um Transfer Control Protocol;

IP 🡪 Internet Protocol;

Para entender melhor a parte de transferência de dados web

<https://www.gta.ufrj.br/grad/03_1/ip-security/paginas/introducao.html>

**JSON e XML: Formatos para Troca de Dados**

**JSON (JavaScript Object Notation)** e **XML (eXtensible Markup Language)** são formatos de texto utilizados para armazenar e transmitir dados estruturados. Embora ambos sirvam para o mesmo propósito, possuem características e aplicações distintas.

Fonte: <https://www.alura.com.br/artigos/o-que-e-json#:~:text=XML%20(Extensible%20Markup%20Language)%20e,h%C3%A1%20diferen%C3%A7as%20na%20sintaxe%20e>