



Pontifícia Universidade Católica de Minas Gerais

Curso: Especialização em Arquitetura de Software Distribuído - Praça da Liberdade
Disciplina: Arquitetura de Front-end
Professor: Rommel Vieira Carneiro

Laboratório: Ambiente e Layout da Aplicação

Instruções gerais

- O trabalho deve ser feito em dupla ou em trio e a entrega deve ser feita no Canvas.
- Os produtos deste laboratório compõem a proposta de arquitetura de front-end a ser entregue no final da disciplina.

Introdução

Neste trabalho, vamos inicialmente preparar o ambiente para a realização das atividades da disciplina e na sequência criar uma proposta de layout da arquitetura para uma aplicação Web utilizando os conceitos apresentados em sala. Isto deverá envolver a definição de componentes para a montagem da estrutura da aplicação e estratégias de responsividade.

Ambiente de Trabalho

Neste laboratório vamos utilizar os seguintes componentes:

- IDE: Visual Studio Code
- Gerenciador de Pacote: NPM
- Bundler: Webpack
- Pre-processador de CSS: Sass
- Gerenciado de códigos fonte (SCM): github

Para montagem do ambiente, abra o Visual Studio Code siga os passos que são detalhados a seguir.

Instalação do Visual Studio Code

Para instalar o Visual Studio Code, será necessário fazer o download a partir do site <https://code.visualstudio.com/> e na sequência proceder com a instalação no sistema operacional

OBSERVAÇÃO: O Visual Studio Code já se encontra instalado nas máquinas do laboratório.

Abra o Visual Studio Code e abra uma pasta para um novo projeto, algo como **c:\ProjetoAFE**.

Instalação do NPM

Caso o NPM não esteja instalado na sua máquina, será necessário fazer o download do Node.JS a partir do site <https://nodejs.org> e na sequência proceder com a instalação no sistema operacional. Utilize a versão LTS que é a versão estável e que apresenta os recursos que vamos precisar. O Node.JS traz consigo o NPM.

Todos os pacotes globais do NPM são instalados em um diretório no seu computador que deve estar no PATH do seu sistema operacional para que você possa executar os aplicativos instalados de qualquer lugar. No Windows, este diretório possui a seguinte estrutura:

```
> C:\Users\[NOME_USUARIO]\AppData\Roaming\npm
```

OBSERVAÇÃO: O NPM já se encontra instalado nas máquinas do laboratório.

Inicialização do projeto com o NPM

Para que o NPM controle todas as dependências do projeto é preciso inicializar a pasta do projeto com o comando abaixo:

```
> npm init -y
```

Isto faz com que seja criado um arquivo **package.json** contendo as configurações para o projeto. Conforme mostrado no quadro que se segue, neste arquivo vamos encontrar as informações descritivas sobre o projeto e vamos ter a relação de pacotes utilizados por este projeto que vão ser criados em uma pasta **node_modules**.

```
{
  "name": "sass",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "deploy": "sass src/sass/main.scss assets/css/main.css"
  },
  "author": "",
  "license": "ISC"
}
```

Quadro 1 – Arquivo package.json

Instalação do Sass

O pré-processador Sass é instalado a partir do NPM e, para tanto, execute o seguinte comando:

```
> npm install -g sass
```

Com isso, o Sass, será instalado no diretório global do NPM. Caso este diretório esteja no PATH, você poderá testar o Sass, executando:

```
> sass --version  
1.17.3 compiled with dart2js 2.2.0
```

Layout da Aplicação

Nossa aplicação deve contar com diversos conceitos que vimos em sala, tais como marcação semântica (HTML) e estrutura de componentes. Para tanto, vamos criar uma estrutura para um site de e-Commerce. Para isso, vamos seguir os seguintes passos:

1. **Identificação de conteúdos necessários** – por meio de uma sessão rápida de *brainstorming*, o grupo deve identificar os elementos visuais da aplicação que vão ser apresentados para o usuário tais como menu, caixa de busca, lista de produtos, slider de destaques, navegação periférica, etc.
2. **Definição do layout base** – o grupo deverá fazer o esboço de pelo menos dois wireframes e distribuir os conteúdos. As telas são:
 - a. tela inicial do site com exibição dos produtos em oferta,
 - b. tela de descrição de produto
3. **Elaboração da marcação** – criação de trechos de código HTML para estrutura (layouts base e outros) e componentes (conteúdos do site).

A figura abaixo apresenta uma ideia da estrutura e componentes da aplicação a ser criada.

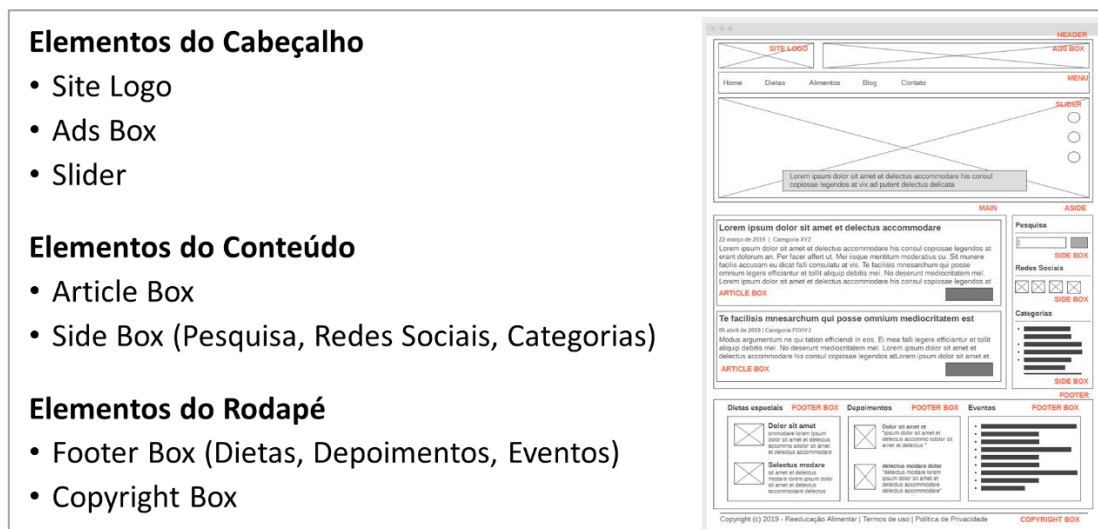


Figura 1 – Projeto de layout de aplicação Web

Uma vez projetada a estrutura da aplicação (layout + componentes), é hora de criar o projeto no ambiente de trabalho, utilizando o Visual Studio Code como ferramenta de codificação.

Codificação da aplicação

Pense inicialmente na sua **estrutura de pastas**, levando em consideração a disposição de páginas HTML, folhas de estilo CSS e arquivos JavaScript. Lembre-se que você deve ter dois espaços no seu projeto, um para o código fonte (src) e outro para a aplicação finalizada (dist), pronta para ser publicada em um site. Com isso, sua aplicação pode se valer de tecnologias como o pré-processador de CSS e outras tecnologias mais recentes do JavaScript. Este código fonte será “transpilado” no ambiente de desenvolvimento antes de ser publicado no ambiente de produção.

Uma vez pensada a estrutura de pastas, é hora de montar os códigos de marcação e definir a estrutura das classes de CSS que serão utilizadas pela aplicação. Comece com o código do **layout base** e, em seguida, crie o código de cada um dos **componentes** identificados para sua aplicação.

Os quadros abaixo apresentam códigos de exemplo para layout base e componentes respectivamente.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
  </head>
  <body>
    <header class="site-header"></header>
    <main class="site-main"></main>
    <aside class="site-aside"></aside>
    <footer class="site-footer"></footer>
  </body>
</html>
```

Quadro 2 - Código de estrutura do layout base

```
<article class="article-box">
  <h2 class="">Tempore totam fugit officiis</h2>
  <p class="detail">
    <span class="date">22/03/2019</span> |
    <span class="category">Vida Saudável</span></p>
  <p class="texto">Lorem ipsum, dolor sit amet consectetur adipisicing elit.
    Ipsa, quia! Dolores nemo fugitdolor incidunt placeat exercitationem
    distinctio. Reiciendis, quod minus neque, tempore ia voluptatibus!</p>
  <p class="footer"><a href="#">Leia mais...</a></p>
</article>
```

Quadro 3 - Código de um componente (article-box)

Na sequência, crie a estrutura de código para as folhas de estilo do CSS. O Quadro abaixo, traz um exemplo de estrutura do CSS a ser definido especificamente para o componente apresentado no código anterior.

```
/****** Componente article-box *****/
$article_title: $color5;
$article_category: $color1;
$article_text: $text_color;

.article-box {
  margin-right: 20px;
  > h2 {
```

```

        color: $article_title;
        margin-bottom: 0;
    }

    > .detail {
        margin: 0;
        font-size: 80%;
    }

    > .detail > .category {
        color: $article_category;
    }

    > .texto {
        color: $article_text;
        margin: 5px 0;
    }

    > .footer {
        margin: 5px 0;
        text-align: right;
    }
}

```

Quadro 4 – Folha de estilo CSS de componente (article-box) em Sass

Automação de tarefas

O NPM oferece alguns recursos para automatizar o ambiente de desenvolvimento e simplificar a vida do desenvolvedor. No arquivo **package.json**, o item **scripts** permite que sejam definidos comandos para que o desenvolvedor possa realizar diversas tarefas de maneira simplificada.

```

"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "deploy": "sass src/sass/main.scss assets/css/main.css"
},

```

Quadro 5 – NPM Scripts

O quadro acima traz um exemplo de código que permite com que o arquivo **main.sass** seja processado a partir da pasta **src** gerando o respectivo código CSS na pasta **assets/css/main.css**. Isso acontece quando o usuário executar o comando:

```
> npm run deploy
```

Agora é hora de automatizar instrumentos de otimização da aplicação com o webpack. Utilize o site para compreender alguns exemplos e implemente isso no seu projeto: <https://github.com/ruanyf/webpack-demos>