UNIVERSIDADE CATÓLICA DE PELOTAS

TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISCIPLINA: PROJETO INTEGRADOR IV-A

PROFESSORES: ALEXANDRA LACKMANN ZIMPECK

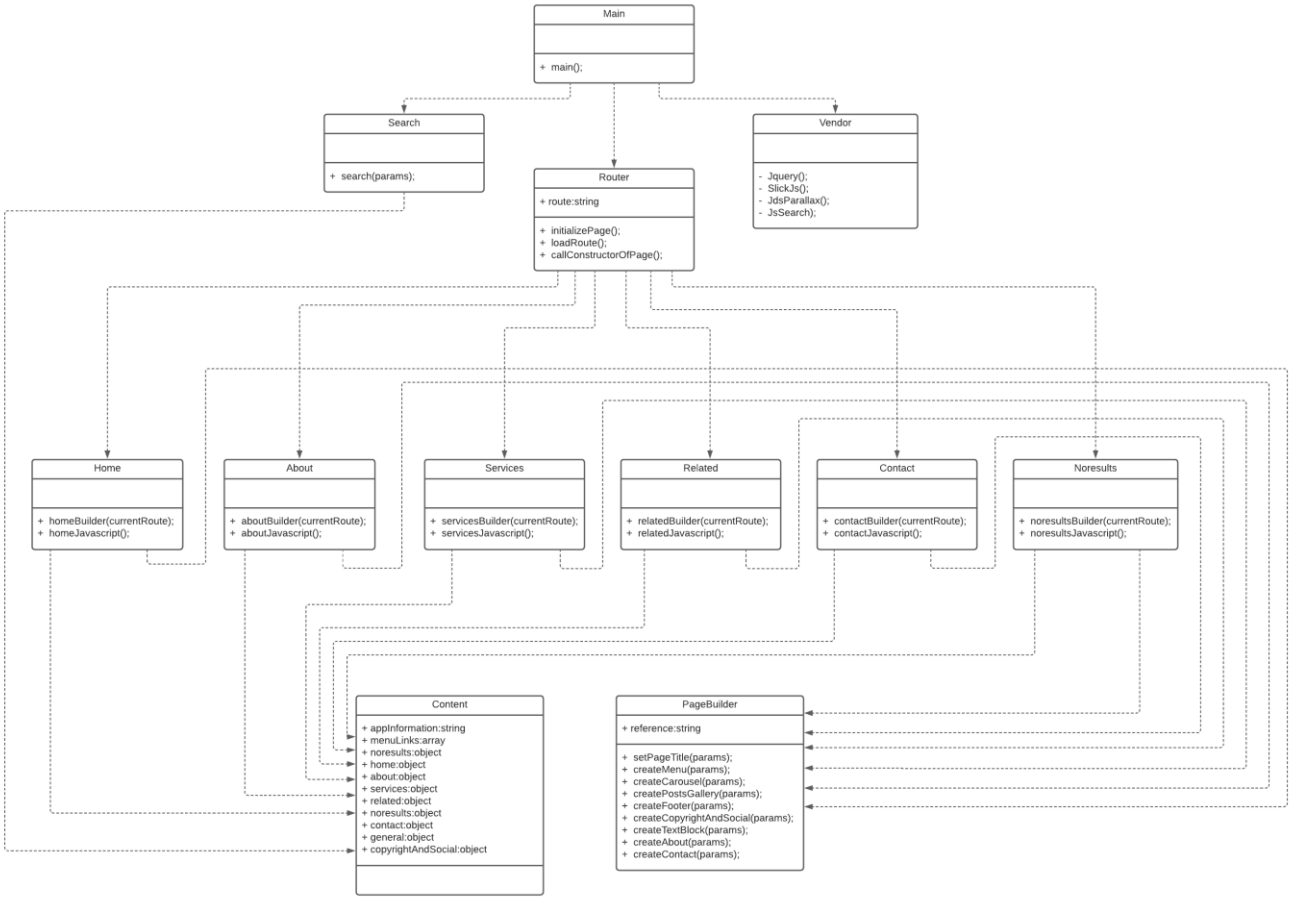
FERNANDA MOTA

RELATÓRIO FINAL

ALUNOS: JULIANO DUARTE SEUS, LUKA VIEIRA IRIBARREM

PELOTAS, OUTUBRO DE 2021

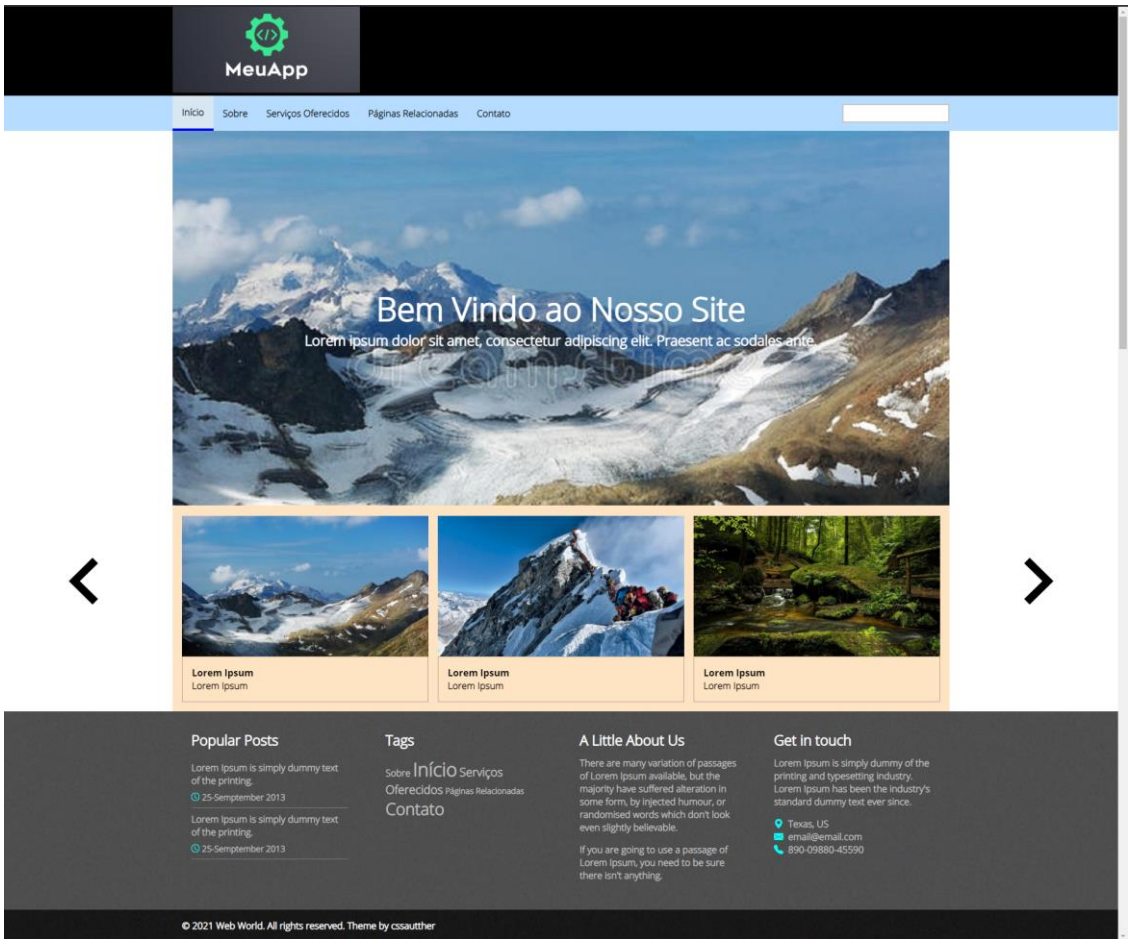
DIAGRAMA DE CLASSES FINAL



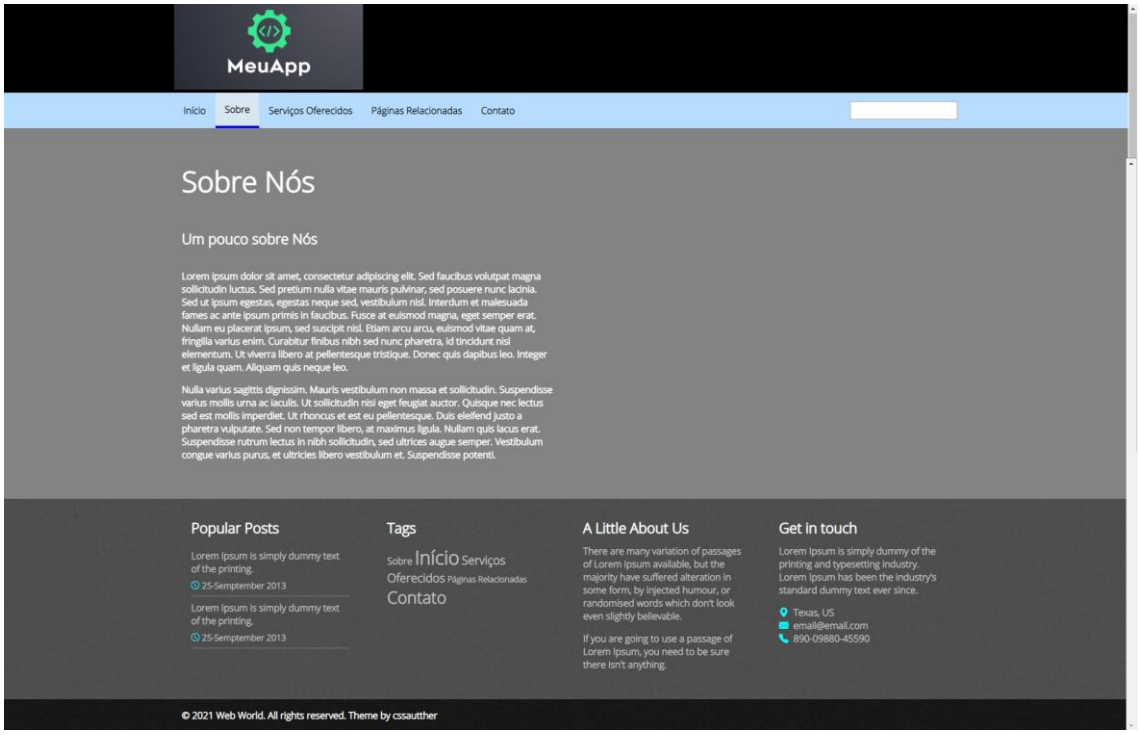
Durante a execução do projeto, revisamos o diagrama original que havíamos proposto e fizemos algumas adaptações resultando no diagrama acima.

PÁGINAS ESTÁTICAS GERADAS

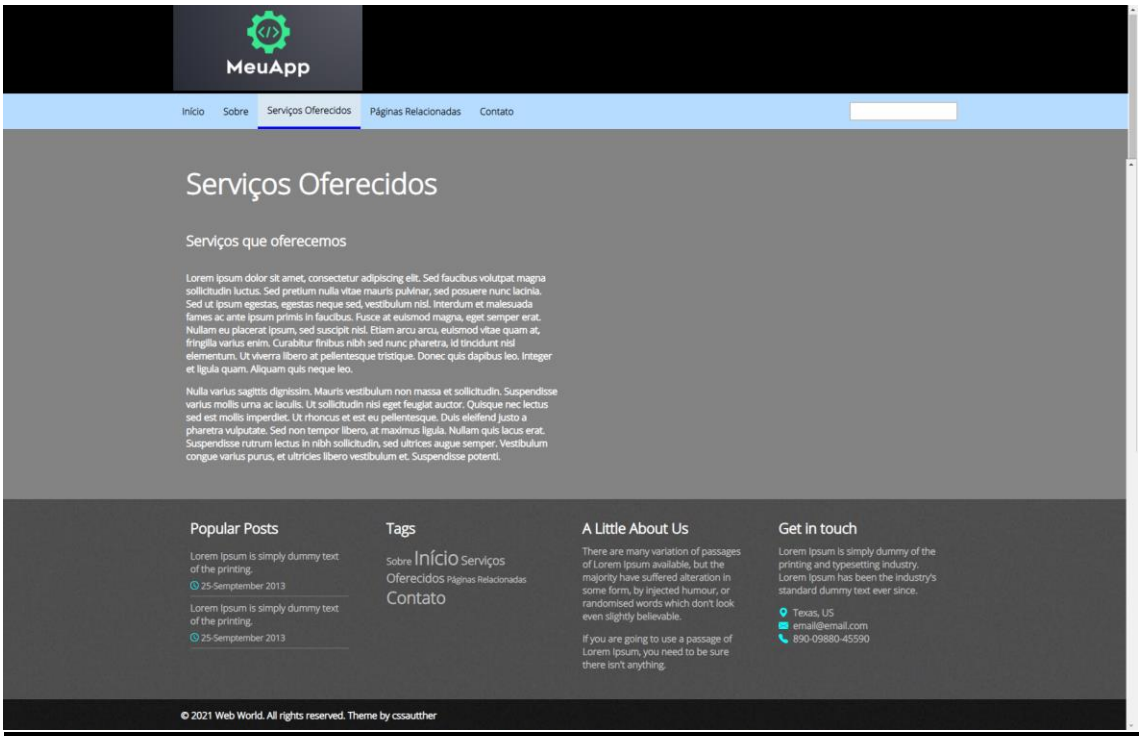
INÍCIO



SOBRE NÓS

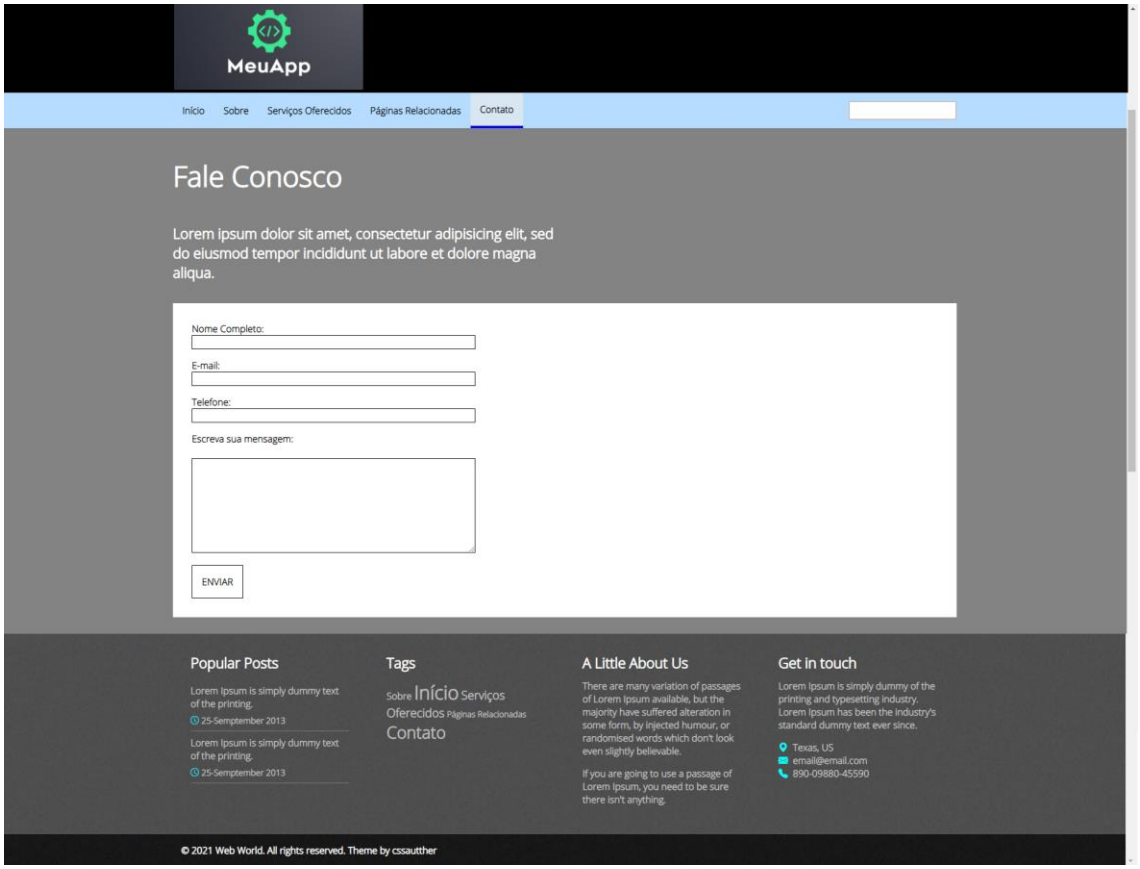


SERVIÇOS



PÁGINAS RELACIONADAS

CONTATO



OBJETIVO DA APLICAÇÂO

O nosso objetivo era gerar as páginas estáticas através de uma classe centralizado em javascript. Utilizando o diagrama de classes mostrado acima conseguimos cumprir este objetivo. A seguir, daremos mais detalhes das classes utilizadas e como esse resultado foi obtido.

A CLASSE CONTENT

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/Content.js

A classe Content é a mais importante do projeto. É nela que ficam registrados todos os textos, links e ícones que aparecem nas páginas. Ela centraliza tudo o que pode ser modificado, de modo que quando queremos editar algum texto das páginas, é nela que editamos. As outras classes utilizam ela para gerar as páginas.

A CLASSE PAGEBUILDER

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/PageBuilder.js

Na classe PageBuilder estão registrados os componentes (código javascript que gera html) que são utilizados nas páginas. Se quisermos editar algum HTML, é nos componentes dela que devemos editar. Ela é muito importante, porque as páginas do projeto são construídas utilizando os seus métodos.

A CLASSE VENDOR

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/vendor/vendor.js

Na classe Vendor estão registradas as bibliotecas necessárias para o funcionamento do projeto. Ela é o primeiro código executado pela aplicação na classe Main.

A CLASSE ROUTER

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/Router.js

A classe Router é a responsável por identificar a rota atual da aplicação e chamar o construtor adequado para gerar o HTML respectivo. Ela faz isso utilizando as classes Noresults, Home, About, Services, Related e Contact, que são as construtoras de cada uma das páginas da aplicação.

AS CLASSES ABOUT, CONTACT, HOME, NORESULTS, RELATED E SERVICES

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/pageBuilders/About.js

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/pageBuilders/Contact.js

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/pageBuilders/Home.js

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/pageBuilders/Noresults.js

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/pageBuilders/Related.js

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/pageBuilders/Services.js

Cada uma dessas classes tem dois métodos sendo eles:

*(Nome da classe)Builder(currentRoute);

*(Nome da classe)Javascript();

Todas elas tem acesso a classe Content e PageBuilder. Desse modo, o método Builder delas chama os componentes da PageBuilder de acordo com o que aquela página está definida para construir. Já o método Javascript contém o Javascript personalizado que vai ser utilizado naquela página. Na página Inicio, por exemplo, o Builder chama os componentes para gerar a página com o carrossel e o Javascript chama o código necessário para ativar o Slick Js.

A CLASSE SEARCH

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/Search.js

A classe Search tem acesso direto a classe Content. O objetivo dessa classe é conter um algoritmo preparado da biblioteca SearchJs de modo a realizar uma busca no conteúdo da Content e retornar a rota em que aquele conteúdo ocorre.

A CLASSE MAIN

https://github.com/JdSeus/JDS-UCPEL-PI4M1/blob/master/resources/js/Main.js

A classe Main utiliza as classes Vendor, Search e Router. Ela executa a Vendor para deixar as bibliotecas preparadas, após isso, ela chama o Router para identificar a página atual e realizar a construção desta. Além disso, ela que coloca o evento de pesquisa no input do header, utilizando a classe Search para analisar a pesquisar, redirecionando de acordo com o resultado da pesquisa.

SOBRE O CSS DO PROJETO

Para gerar o CSS do projeto foi utilizado SCSS na estrutura de pastas recomendada pela documentação da ferramenta. Também minificamos o CSS através do site https://cssminifier.com/.

MAIS DETALHES SOBRE O JAVASCRIPT DO PROJETO

Como utilizamos alguns recursos do ECMAScript 6 (como os módulos nativos), nós compilamos e minificamos o Javacript utilizando Webpack, de

modo a diminuir o peso, colocar tudo em apenas um arquivo e aumentar a compatibilidade entre browsers.

CÓDIGO FONTE

O código fonte do Projeto está em anexo.

CONCLUSÃO

Foi uma experiência bem interessante utilizar o Javascript com um paradigma mais orientado a objetos. A ideia de centralizar os conteúdos do site em uma classe também se mostrou boa para a facilitação de manutenção de conteúdo. Certamente levaremos em consideração estes resultados que obtivemos para projetos futuros.