

Criação de um Site com Conteúdo Estático

Com bons recursos para gerenciamento do conteúdo

Após ter criado o site usando o Bootstrap, que tem ótimos recursos, decidi agora criar outro site, com o mesmo template (aparência) e as mesmas funcionalidades mas apenas com recursos mais simples, sem o bootstrap, para que possa ter um controle completo sobre o mesmo.

Objetivo

O objetivo é criar uma estrutura de site tão prático quanto usando um CMS, mas somente com conteúdo estático. Usando boas práticas e ferramentas para poder abrigar este site em servidores gratuitos como o GitHub.

Controle

Neste projeto preciso de controle completo sobre a estrutura do site, por isso usarei apenas componentes bem simples ou apenas código simples em HTML, CSS e JavaScript mas ainda assim também simples de entender e alterar.

Solução Adotada

- Criar um arquivo cabeçalho.html, contendo o head e o menu
- Criar um pequeno script com o método load da jQuery que importa o cabeçalho.html
- Adicionar este pequeno script em cada arquivo de conteúdo

Recursos utilizados:

- Usei um template com layout moderno e agradável (O Visualize criado pela Templated - <https://templated.co/visualize>)
- O layout do template é responsivo
- Como grid usei SimpleGrid - <https://thisisdallas.github.io/Simple-Grid/>
- Menu dropdown multinível e fixo mas agora na lateral esquerda
<http://stackoverflow.com/questions/8141513/how-to-create-a-css-only-vertical-drop-down-menu>
Além de funcional tem um código bem simples de entender e fácil de alterar.
- Editor HTML FCKEditor para formatar o conteúdo e criar as páginas (<http://fckeditor.com>)
- Editor Ace, que ajuda a corrigir erros no HTML, JavaScript, CSS e muito mais
(<https://github.com/ajaxorg/ace-builds>)
Evitarei usar a ferramenta acima procurando indentar rigorosamente as tags para que possa identificá-las com facilidade
- O path de todos os includes deve ser o path absoluto, para que seja visto por todos os subdiretórios
- Os includes dos javascript são inseridos ao final de cada arquivo de conteúdo, visto ser pouco conteúdo, mas pode ser criado um outro arquivo e importado com o load. Lembrando que nem sempre isso é possível. Alguns componentes exigem que o JavaScript esteja no início para funcionar. Mas neste projeto não foi necessário nenhum arquivo de javascript, a não ser o da jquery, por conta do método load.

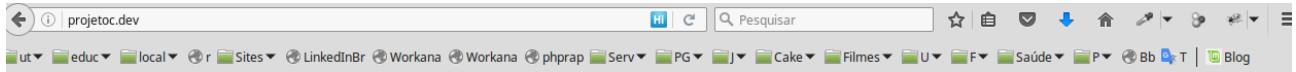
Domínio Virtual

Começamos criando um domínio virtual para nosso projeto, que chamarei de projetoc.dev que fica na pasta /var/www/html/projetoc

Baixei o template e descompactei na pasta do projetoc. Este template vem apenas com uma pasta assets (css, js e imgs), uma pasta images (imagens do portfolio) e um arquivo index.html e um LICENSE.txt, da licença Creative Commons.

Chamei pelo navegador para verificar o que deixo e o que removo.

Veja ele original



This is Visualize, a responsive site template designed by [TEMPLATED](#) and released for free under the Creative Commons License.

- [Twitter](#)
- [Facebook](#)
- [Instagram](#)
- [500px](#)
- [Email](#)

[Lorem ipsum dolor sit amet](#)

[Lorem ipsum dolor sit amet](#)

[Lorem ipsum dolor sit amet](#)

[Lorem ipsum dolor sit amet](#)

[Lorem ipsum dolor sit amet](#)

[Lorem ipsum dolor sit amet](#)

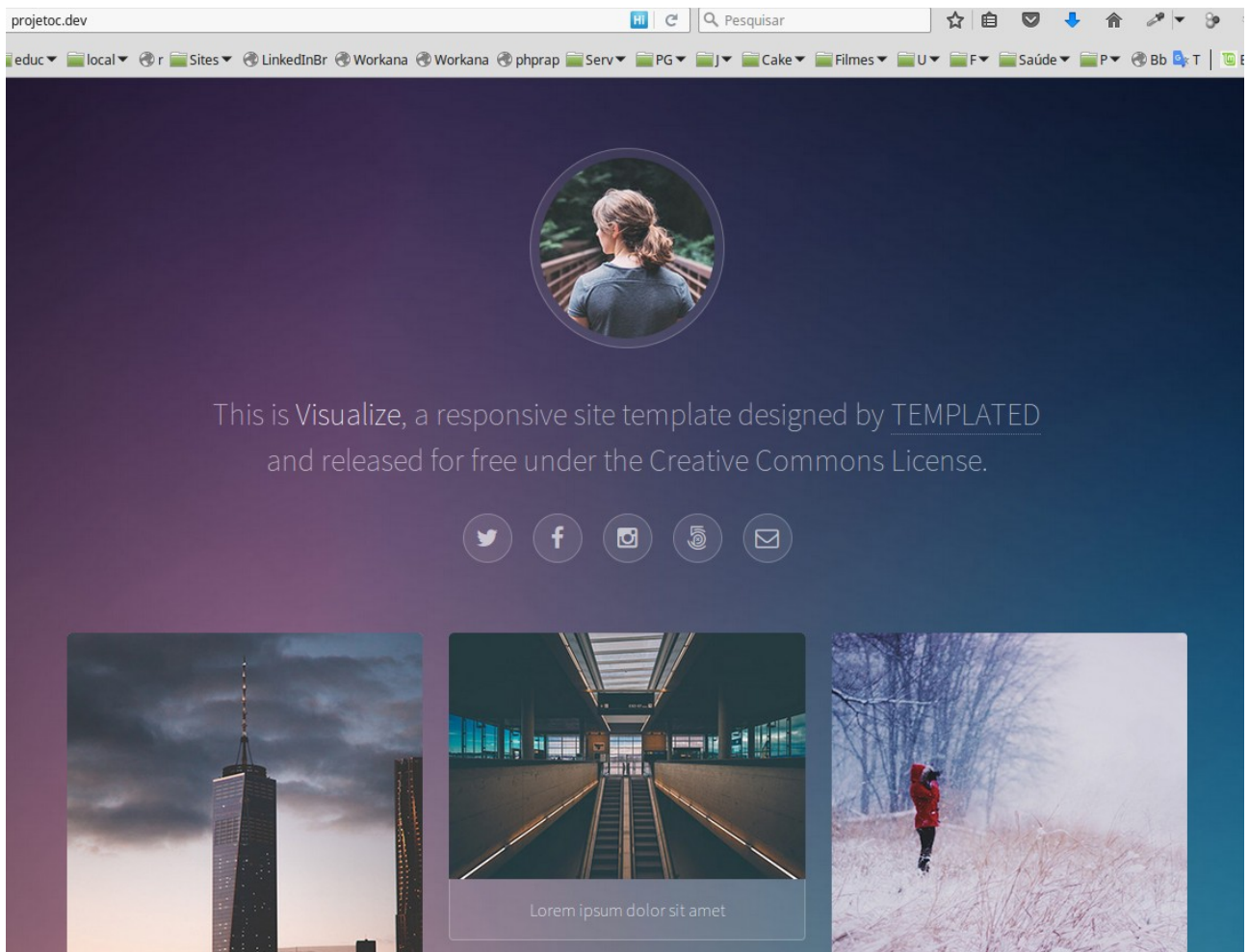
[Lorem ipsum dolor sit amet](#)

© Untitled. All rights reserved. Design: [TEMPLATED](#). Demo Images: [Unsplash](#).

Por que isso?

Por que estou usando no linux em meu desktop e ainda não encontrei uma forma de setar as permissões de forma automática. Como descompactei o conteúdo para a pasta no /var, agora preciso executar um script que criei para ajustar as permissões.

Veja:



Agora vamos começar.

Qual é a ideia deste template?

Veja que não existem menus. Parece que quer apenas mostrar um portfolio juntamente com uma foto e contatos.

Vou trocar a foto por uma minha, mas também meio de perfil, visto que não estou querendo vender minha imagem. O colega André me ajudou aqui, criando esta foto. Valeu novamente André.

Vou trocar as imagens do postfolio por ícones/fontes tipo os dos contatos (usando as fontes awesome (<http://fontawesome.io/icons/>)). Adicionando links e uma pequena descrição. Ao invés de usar menus tradicionais usarei estas novas estruturas para cada área de conteúdo do meu site.

Interessante é que o arquivo font-awesome.min.css é importado no início do main.css usando:

```
@import url(font-awesome.min.css);
```

Aí também importa fontes:

```
@import url("http://fonts.googleapis.com/css?family=Source+Sans+Pro:200");
```

Moverei os ícones de contato para o rodapé.

O que trocar?

Foto

A foto por uma minha.

Reduzirei um pouco. Isso será feito reduzindo a foto em 70% e no assets/css/main.css:

```
#header .avatar img {  
    border-radius: 100%;  
    display: block;  
    width: 7em; /* Originalmente era 10 */  
}
```

Posso deixar a foto no tamanho original, que talvez tenha uma melhor qualidade e reduzir apenas no CSS.

Frase de Boas Vindas

Olá!

Seja sinceramente muito bem vindo(a)!

Fique inteiramente à vontade e escolha um dos assuntos abaixo para tem acesso.

Qualquer crítica ou sugestão, por favor, me contacte.

Scripts

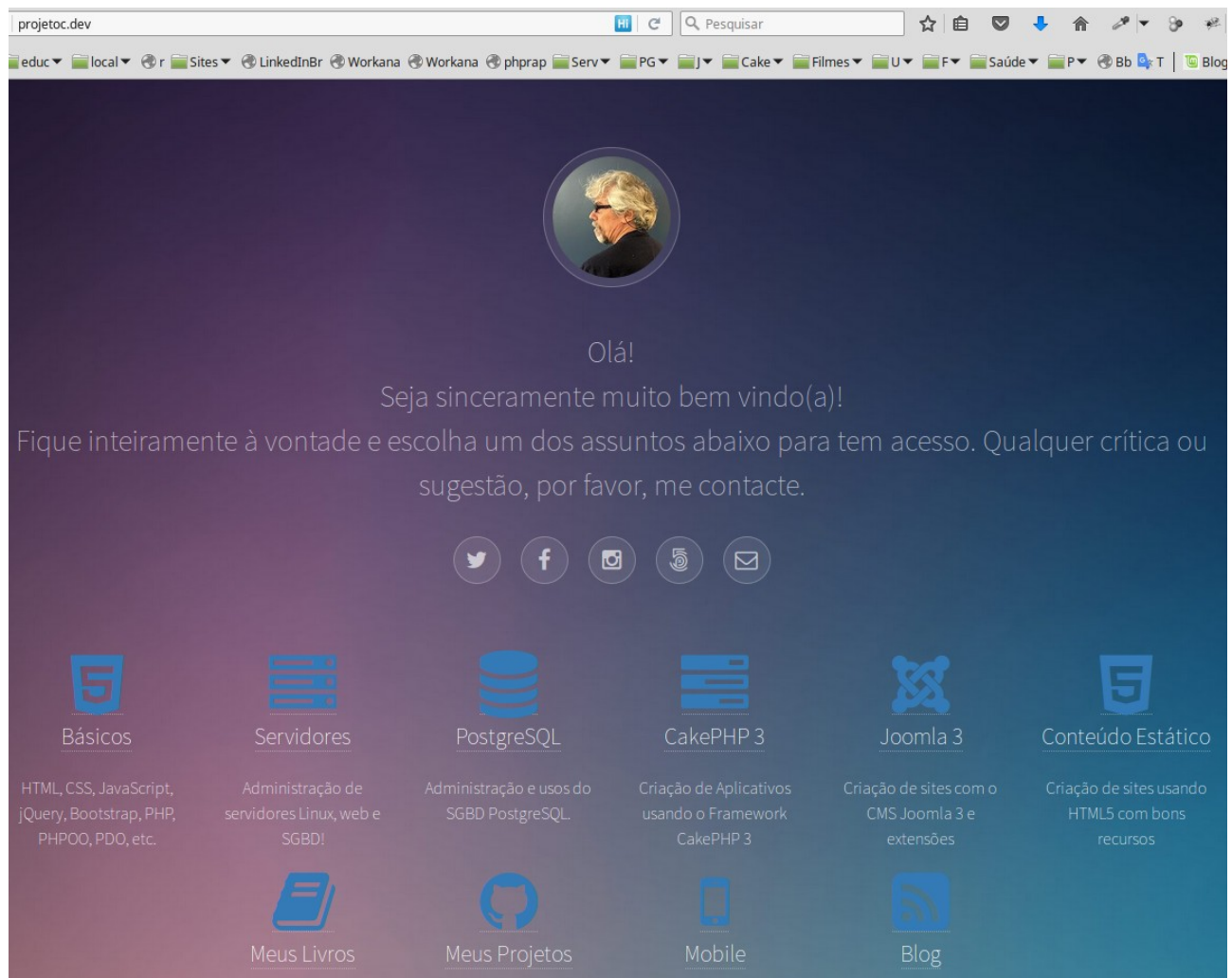
Existia includes de 4 scripts js no rodapé que removi, pois não eram utilizados.

As fotos do portfolio por ícones/fontes, com links e pequenas descrições.

E troquei o código do portfólio por um para ícones/fontes para cada um dos antigos itens de menu do site antigo.

Para isso criei uma grid usando o SimpleGrid. Apenas copiei o código que usei no site criado anteriormente, mas não ficava como eu queria. Não ficava como antes. Ficava com os ícones brancos e não ficava centralizado. Centralizei nas divs das duas linhas e pesquisando sobre a cor encontrei novamente no “stackoverflow” adicionei estilo em cada linha para mudar a cor.

Veja agora:



Removi as fotos existentes da pasta images, deixando apenas bg.jpg, que é usada no background e trocando avatar pela minha foto.

A bg.jpg é usada no main.css, como background-image do body.

Sem o bootstrap os ícones/fontes não ficaram azuis, mas brancos. Então fiz uma rápida pesquisa e adicionei ao final da tag <i>:
style="color:#337AB7"

E ficou igual, pois capturei a cor usando a excelente extensão ColorZilla para o Firefox.

Avatar

Subi mais a imagem avatar.
Para isso alterei o main.css

```
#header {  
    padding: 1em 0 2em 0; /* Era 4 onde tem 1.  
}
```

Como adiviquei? Fui procurando top, padding e parecidos. Depois voltei para 4, pois achei melhor.

Seção de Ícones do Conteúdo

O layout desta seção foi criado pesquisando pela grid do bootgrap. Agora adaptado para o SimpleGrid.

Criei duas linhas, sendo a primeira com 6 colunas de largura 2 e a segunda com apenas 5.

Observe que adicionei links também nas imagens, para ser mais amigável.

Contatos

Movi os contatos para a região do rodapé, pois tudo indica que as pessoas chegam no meu site, mais pelo conteúdo que para me contatar. Pensando assim deveria mover a foto para abaixo dos contatos, mas vou deixar acima por questão de estética.

Adicionei alguns `
` para espaçar mais as seções.

Enxugar CSS

O arquivo main.css tem muita coisa não utilizada no site e irei remover agora.

Apenas citarei as seções (estão comentadas):

- grid (enorme)
- form
- box
- image
- list (boa parte)
- table
- buttom
- thumbnails
- Poptrox

E mais algumas pequenas remoções, sempre tendo o cuidado de atualizar o site e verificar se não fez falta.

Deixar isso gera dois problemas: gordura desnecessária (que degrada o desempenho) e pode entrar em conflito com novos CSS futuramente adicionados.

O arquivo original tinha 43,9 kb e ficou com apenas 7kb. Isso significa problemas a menos, mesmo que para isso eu tenha gastado algum tempo, mas que não foi grande.

Os arquivos de CSS do bootstrap e o de fontes não irei otimizar. Poderia também, para isso usando o fonte e não o mínimo. Ao final compacta-se e usa-se a versão mínima no site.

Excluir JavaScripts

Nenhum dos scripts em javascript que acompanham o template Visualize são usados, então eu os removi inclusive seus includes da index.html.

Somente a jquery foimantida mas não na index, somente nas outras páginas por conta do método load.

Adicionar Conteúdo

Agora é a hora de adicionar todo o meu conteúdo e customizar para que fique com o mesmo template.

Import do HTML

Criarei um arquivo cabecalho.html para cada área de conteúdo contendo o head, juntamente com os includes de CSS e também com o menu da respectiva seção.

Também usarei um pequeno script em JavaScript que usa o método load da jQuery que será adicionado ao início de cada página com conteúdo para importar o cabecalho.html da respectiva seção.

Assim:

- script com o load para o cabecalho.html
- arquivo html com o conteúdo

```
<script src="/assets/js/jquery.min.js"></script>
<script>
  $(function(){
    $('#cabecalho').load("./cabecalho_basicos.html");
  });
</script>
<div id="cabecalho"></div>
```

Adicionar a Primeira seção

Adicionar Conhecimentos básicos.

Para isso crio uma sub pasta chamada **basicos**.

Dentro dela crio um arquivo chamado cabecalho.html e outro chamado index.html.

Cada nova página de conteúdo é composta de

cabecalho.html
conteudo.html (este arquivo em o script com load no início)

No cabecalho.html entro com o head, os includes css e o menu.

No index.html entro no início o script que importará o cabecalho.html e abaixo entro o contrúdo propriamente dito.

Quando fui inserir um ícone/fonte do HTML5 na index de Conhecimentos Básicos, ele não apareceu. Virei, mexi e suspeitei que o menu.css o estava escondendo. Então mudei o include do menu.css para cima do bootstrap e apareceu. Precisamos ficar atentos com isto.

Menu DropDown Multi Nível e Fixo na Margem Lateral Esquerda

Foi ótimo, pois as páginas de conteúdo, para facilitar a leitura eu reduzi sua largura, ficando um espaço livre à esquerda e à direita, pois deixei centralizado (usei 'margin 0 auto;' no body).

Gostei deste menu pois é muito simples. O CSS altera o comportamento das tags e não preciso ficar inserindo classes nem ids, apenas uso as tags ul e li. A estrutura do menu fica bem mais simples que outras soluções, mesmo que a do Bootstrap.

Adicionando Conteúdo

Para adicionar o conteúdo faço assim: pegando como exemplo a seção **Conhecimentos Básicos** do meu site:

- Criei o sub diretório **basicos**;
- Criei os arquivos **cabecalho_basicos.html** e o **index.html**. No cabecalho coloco o head e o menu. No index.html coloco um texto introdutório sobre o assunto.
- Esta seção é sobre HTML, CSS, JavaScript, jQuery, etc.
- Então se quiser falar pouco sobre cada um destes assuntos eu apenas crio um arquivo .html sobre cada um e o chamo no menu. Se quiser falar muito eu crio um sub diretório para cada um e dentro dele coloco arquivos para cada assunto.

Ferramentas

Editor HTML FCKEditor

Supondo que queira criar um arquivo chamado xdk.html.

Abro o FCKEditor para criar o conteúdo. É um editor muito bom, onde formato fontes, com tamanho, negrito, itálico, alinhamento texto, insiro imagens, etc.

Quando o conteúdo está pronto clico no botão Código-Fonte, seleciono tudo, copio, então abro o gedit e colo todo o conteúdo e salvo como xdk.html.

Depois disso abro o xdk.html no gedit e adiciono em seu início o código do script com o método load, citado acima.

Salvo e está pronto meu arquivo xdk.html.

Criando um Cabeçalho

Para criar um arquivo de cabeçalho uso apenas um editor simples como o gedit, pois ele contém apenas código HTML, CSS e JavaScript.