

## Importação de arquivos

### Transcrição

Quando queremos editar uma parte específica do arquivo `.scss`, por exemplo o `planos`, precisamos procurar o trecho em que ele está. Isso não é tão prático, mesmo que usemos o "CTRL + F". Mesmo com o arquivo comentado, seu tamanho atrapalha um pouco na hora de encontrarmos o que queremos.

Para deixar tudo mais organizado, podemos isolar cada parte em um arquivo separado, dentro da pasta "css". Assim, deixaremos as variáveis, os *mixins*, o `header`, o `destaque`, o `sobre`, o `planos`, o `blog`, o `contato` e o `footer` em seus próprios arquivos, que lhes serão homônimos. Usando o "CTRL + X", recortamos o trecho de código referente a cada parte e colamos uma a uma em um arquivo novo, criado com "CTRL + N".

Depois de terminado esse processo, o arquivo `estilos.scss` estará vazio. Se quando este arquivo for salvo, o `.css` não atualizar, é bom ir ao terminal e verificar o que está acontecendo. A essa altura ele já está bem poluído com informações antigas, então é válido usar o comando `clear` para limpar a tela.

Repetimos agora o comando para que o arquivo `.css` volte a observar mudanças no `.scss`:

```
sass --watch estilos.scss:estilos.css
```

E o terminal novamente deve nos responder com:

```
>>> Sass is watching for changes. Press CTRL-C to stop.
```

E quando voltamos ao editor de texto, nosso arquivo `.css` nos mostra apenas:

```
/* sourceMappingURL=estilos.css.map */
```

Todo o resto do código está ausente, o que é um problema. Acabamos de pedir no terminal que o `.css` observasse o `.scss` – que está vazio! Como o `.css` não é adivinho, precisamos importar todos os novos arquivos que criamos para o `.scss`, na mesma ordem que apareciam originalmente. O comando para importação é `@import`, seguido do nome dos arquivos entre aspas simples. Uma das vantagens do Sass é não ser necessário colocar a extensão do arquivo quando ela for `.scss`, que pode gerar erros por problemas de digitação, como `".scss"` ou `"scssx"`. Caso esses erros aconteçam, o `.css` nos avisa que há algo errado com a seguinte mensagem:

```
Error: File to import not found or unreadable: destaque.scssx
```

Mas como o Sass não exige que coloquemos as extensões, essa é uma preocupação a menos. Com isso em mente, começamos a importação pelos arquivos de variáveis e *mixins*:

```
@import 'variaveis';  
@import 'mixins';
```

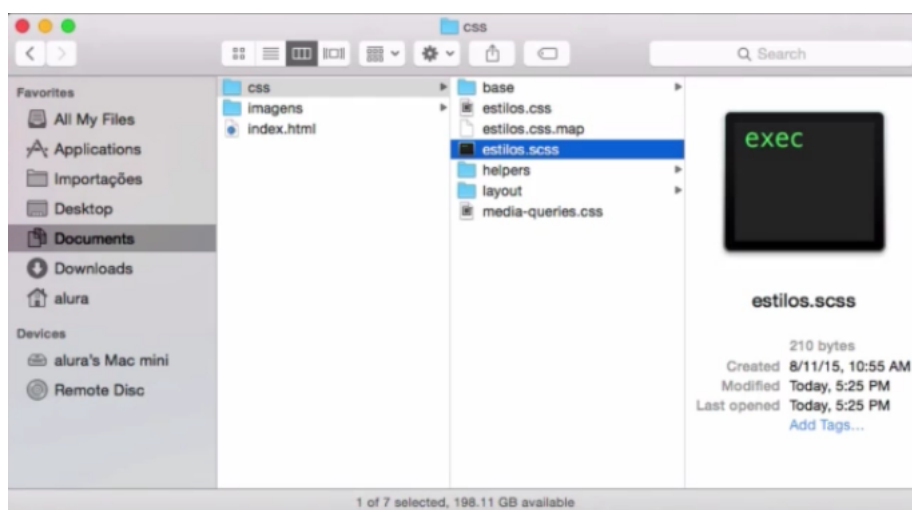
Nesse momento, quando salvarmos, não devemos nos alarmar por nada de novo aparecer no arquivo `.css`. Isso acontece porque esses dois arquivos que importamos não contêm CSS de fato, apenas configurações do Sass. Podemos prosseguirmos com a importação e adicionar o restante dos arquivos. É interessante lembrar que, caso queiramos mudar a ordem de algum dos elementos, mudar a ordem deles na importação é um jeito prático de fazer. Mas, nesse caso, manteremos a ordem original:

```
@import 'variaveis';
@import 'mixins';

@import 'geral';
@import 'header';
@import 'destaque';
@import 'sobre';
@import 'planos';
@import 'blog';
@import 'contato';
@import 'footer';
```

Salvando o arquivo, o `.css` já estará completo e atualizado.

A organização pode ser ainda mais otimizada. Ao abrir a pasta "css", vemos que os arquivos de layout que acabamos de criar estão um pouco soltos ali, dando uma aparência bagunçada à pasta.

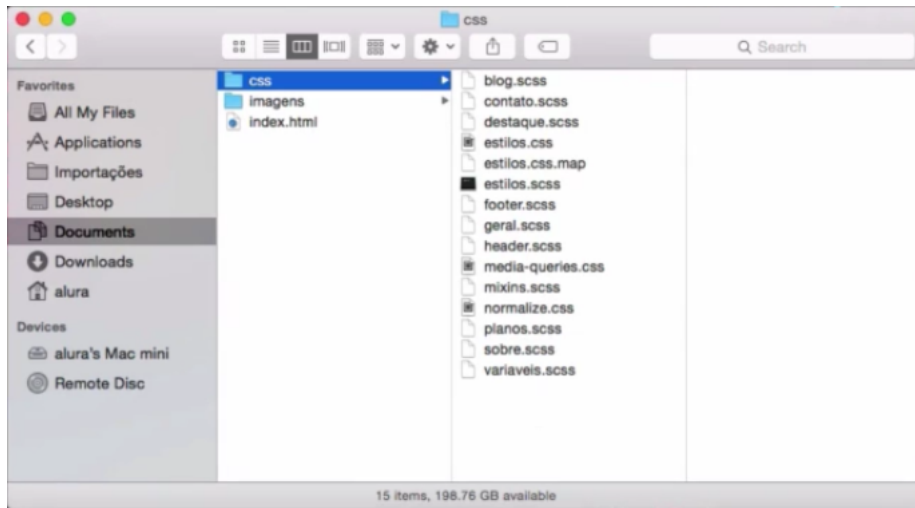


Criar uma pasta "layout" para organizar esses arquivos é a solução. Mas ainda temos o arquivo de *mixins* e de variáveis, que não são exatamente parte do layout. Eles são meramente arquivos ajudantes, então podem ficar em uma pasta separada, que chamaremos de "helpers". Criaremos ainda uma pasta chamada "base", na qual será colocado o arquivo "normalize".

Ao terminarmos essa organização, o `.css` apresenta uma mensagem de erro:

```
Error: File to import not found or unreadable: variaveis.
on line 1 of estilos.scss
```

Mesmo quando adicionamos a extensão e deixamos `variaveis.scss`, o erro persiste. Analisemos novamente a nossa pasta "css" para descobrir o problema:



Quando separamos o arquivo `.scss` dos outros, ele não os encontra. Temos que especificar dentro dele o caminho de importação dos arquivos. O código ficará assim:

```
@import 'helpers/variaveis';
@import 'helpers/mixins';

@import 'layout/geral';
@import 'layout/header';
@import 'layout/destaque';
@import 'layout/sobre';
@import 'layout/planos';
@import 'layout/blog';
@import 'layout/contato';
@import 'layout/footer';
```

Salvando o arquivo, a compilação volta a funcionar normalmente.

É interessante separar cada parte do site em um arquivo diferente, e é possível ir ainda mais fundo e separar cada componente dele, usando o mesmo processo e criando uma pasta "components". Até a próxima!