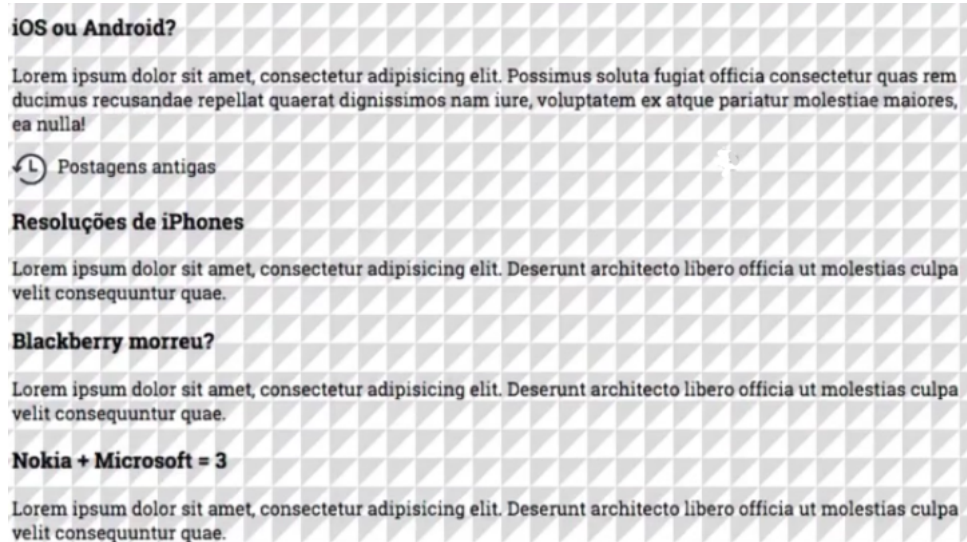


## Introdução a media queries

### Transcrição

*Media query* é um recurso que diz respeito a design responsivo, tornando o site adaptável a diferentes tipos de tela. Por exemplo, ao diminuir a janela do navegador, a seção Blog do site da Apeperia passa de duas para uma coluna:



Esse *media query* específico funciona quando a largura do site passa a ter 980px ou menos. Descobrimos isso indo ao arquivo separado `media-queries.css`.

```
@media (max-width: 980px) {  
  .container  
    width: 90%;  
}
```

Uma prática comum é colocar o *media query* direto no CSS. Mas como o nosso `.css` é o arquivo compilado, essa mudança seria sobrescrita na próxima atualização do arquivo. Um *media query* sobrepõe regras a partir de uma mudança que acontece, então o ideal é que os comandos de *media query* apareçam no final do `.css`. Afinal, é preciso que já haja regras para que elas sejam sobrescritas.

Para fazer isso, podemos usar a função `@import`, dentro do `.scss`. Como o `@import` segue a ordem em que colocamos os arquivos, o código ficará assim:

```
@import 'helpers/variaveis';  
@import 'helpers/mixins';  
  
@import 'base/normalize';  
@import 'layout/geral';  
@import 'layout/header';  
@import 'layout/destaque';  
@import 'layout/sobre';  
@import 'layout/planos';  
@import 'layout/blog';  
@import 'layout/contato';  
@import 'layout/footer';  
@import 'media-queries.css';
```

Ao salvarmos o arquivo, vemos que o código referente ao `media-queries.css` foi parar no topo do arquivo usando o `@import` do CSS, o que não é vantajoso. Passamos por algo similar com o arquivo `normalize`, que também era `.css`. Seguiremos o mesmo procedimento: renomearemos a extensão do arquivo (após fechá-lo, para não gerar conflitos) para `.scss`. Agora não precisaremos mais manter a extensão dentro do código, pois quando não há extensão definida, o Sass já presume que é `.scss`.

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

@import 'base/normalize';
@import 'layout/geral';
@import 'layout/header';
@import 'layout/destaque';
@import 'layout/sobre';
@import 'layout/planos';
@import 'layout/blog';
@import 'layout/contato';
@import 'layout/footer';
@import 'media-queries';
```

Agora sim temos o `media-queries` no final do arquivo compilado. Analisando as regras do `media-queries`, vemos que, justamente por estar no final do arquivo, elas estão distantes dos seletores a que se referem.

```
@media (max-width: 980px) {
  .container {
    width: 90%;
  }

  header {
    height: auto;
  }

  header .container {
    position: static;
  }

  header h1 {
    max-width: 50%;
    margin: 0 auto;
  }

  header h1 img {
    max-width: 100%;
    margin: .5em auto;
    display: block;
  }

  .menu-opcoes {
    position: static;
    display: block;
    text-align: center;
  }
}
```

```

    .menu-opcoes li {
        margin: .4em;
    }

    .destaque {
        font-size: 14px;
        height: 250px;
    }

    .menu-opcoes,
    .menu-opcoes li,
    .post-destaque,
    .posts-antigos,
    form {
        width: auto;
    }
}

```

Por exemplo, seria interessante se a regra referente a `header .container` ficasse próxima do `header .container`, que está assim:

```

header .container {
    position: relative
}

```

O Sass permite a criação de um *media query* interno, direto no seletor. Faremos isso no arquivo `header.scss`, copiando a condição de ativação do *media query* dentro do seletor em questão, que é o `header .container`. E dentro disso, o que deve mudar quando essa condição acontece, que é o posicionamento se tornar estático.

```

header .container{
    position: relative;

    @media (max-width: 980px) {
        position: static
    }
}

```

No arquivo compilado, após salvarmos, o *media query* também ficou interno. Isso ajuda a nos localizarmos dentro do código, e é recomendável indentá-lo mesmo no `.css` para melhorar a visualização. Já que vamos continuar internalizando os *media queries*, podemos tirar o comando `@include` para esse arquivo no `.scss`:

```

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

@import 'base/normalize';
@import 'layout/geral';
@import 'layout/header';
@import 'layout/destaque';
@import 'layout/sobre';
@import 'layout/planos';
@import 'layout/blog';

```

```
@import 'layout/contato';  
@import 'layout/footer';  
@import 'media-queries';
```

Dentro do arquivo `media-queries.css`, podemos deletar o código que acabamos de internalizar.

```
@media (max-width: 980px) {  
  .container {  
    width: 90%;  
  }  
  
  header {  
    height: auto;  
  }  
  
  header .container {  
    position: static;  
  }  
  
  header h1 {  
    max-width: 50%;  
    margin: 0 auto;  
  }  
  
  header h1 img {  
    max-width: 100%;  
    margin: .5em auto;  
    display: block;  
  }  
  
  .menu-opcoes {  
    position: static;  
    display: block;  
    text-align: center;  
  }  
  
  .menu-opcoes li {  
    margin: .4em;  
  }  
  
  .destaque {  
    font-size: 14px;  
    height: 250px;  
  }  
  
  .menu-opcoes,  
  .menu-opcoes li,  
  .post-destaque,  
  .posts-antigos,  
  form {  
    width: auto;  
  }  
}
```

Seguindo a ordem que aparece no arquivo, vamos inserir o código no `container`, que está dentro do arquivo `geral`.

```
.container {  
  width: 940px;  
  margin: 0 auto;  
  
  @media (max-width: 980px) {  
    .container {  
      width: 90%;  
    }  
  }  
}
```

Ao salvarmos, dá erro na compilação:

```
Error: Invalid CSS after "{": expected "}"
```

Ou seja, ele avisa que está faltando uma chave fechada. Revendo o código acima, a chave que está faltando é a que fecha o `@media`. Corrigindo:

```
.container {  
  width: 940px;  
  margin: 0 auto;  
  
  @media (max-width: 980px) {  
    .container {  
      width: 90%;  
    }  
  }  
}
```

O erro agora está corrigido e a compilação volta a funcionar normalmente. A vantagem de um *media query* dentro de seu seletor é a facilidade que traz para a manutenção, pois fica possível ver de uma vez o parâmetro original e qual será seu novo valor. Também é possível acrescentar outras mudanças, como no nosso exemplo, em que acrescentaremos a margem do `container`:

```
.container {  
  width: 940px;  
  margin: 0 auto;  
  
  @media (max-width: 980px) {  
    .container {  
      width: 90%;  
      margin: 10px;  
    }  
  }  
}
```

Na sequência do arquivo de *media query*, temos o `header`. Usando "CTRL + X" para já o tirarmos do arquivo e internalizá-lo no `header`:

```

header {
  border-top: 5px solid $cor-padrao;
  background: rgba($cor-auxiliar, 0.8);
  height: 90px;
  width: 100%;
  position: absolute;

  @media (max-width: 980px) {
    header {
      height: auto;
    }
  }
}

```

E quando vamos conferir o arquivo compilado:

```

header {
  border-top: 5px solid #c24e4b;
  background: rgba(30, 44, 53), 0.8);
  height: 90px;
  width: 100%;
  position: absolute; }

@media (max-width: 980px) {
  header header {
    height: auto; } }

```

Note que dentro do `@media` o `header` se repete. A vantagem de internalizar um *media query* é justamente não precisar repetir o seletor. Mas foi isso que fizemos acidentalmente no `.scss`. Devemos corrigir o código nesse arquivo:

```

header {
  border-top: 5px solid $cor-padrao;
  background: rgba($cor-auxiliar, 0.8);
  height: 90px;
  width: 100%;
  position: absolute;

  @media (max-width: 980px) {
    height: auto;
  }
}

```

Agora o arquivo compilado deixou de repetir o seletor:

```

header {
  border-top: 5px solid #c24e4b;
  background: rgba(30, 44, 53), 0.8);
  height: 90px;
  width: 100%;
  position: absolute; }

@media (max-width: 980px) {

```

```
header {  
  height: auto; } }
```

Resolvido esse item, passamos aos próximos, que se referem a `header h1`. Mas, revendo o código do `header`, não encontramos esse seletor.

```
header {  
  border-top: 5px solid $cor-padrao;  
  background: rgba($cor-auxiliar, 0.8);  
  height: 90px;  
  width: 100%;  
  position: absolute;  
  
  @media (max-width: 980px) {  
    height: auto;  
  }  
}
```

Como `header h1` será um *media query* interno do `header` de qualquer forma, podemos simplesmente criá-lo.

```
header {  
  border-top: 5px solid $cor-padrao;  
  background: rgba($cor-auxiliar, 0.8);  
  height: 90px;  
  width: 100%;  
  position: absolute;  
  
  @media (max-width: 980px) {  
    height: auto;  
  
    h1 {  
      max-width: 50%;  
      margin: 0 auto;  
    }  
  }  
}
```

Para o seletor seguinte, `header h1 img`, usaremos dois recursos simultaneamente: *media query* interno e aninhamento (pois deverá estar dentro do `header h1`).

```
header {  
  border-top: 5px solid $cor-padrao;  
  background: rgba($cor-auxiliar, 0.8);  
  height: 90px;  
  width: 100%;  
  position: absolute;  
  
  @media (max-width: 980px) {  
    height: auto;  
  
    h1 {  
      max-width: 50%;  
      margin: 0 auto;  
  
      img {  
        width: 100%;  
        height: auto;  
      }  
    }  
  }  
}
```

```

    img {
      max-width: 100%;
      margin: .5em auto;
      display: block;
    }
  }
}

```

Para não nos perdermos novamente com a quantidade de chaves, podemos inserir comentários do Sass (feitos com `//`, que não aparecerão no arquivo compilado) que as identifiquem.

```

header {
  border-top: 5px solid $cor-padrao;
  background: rgba($cor-auxiliar, 0.8);
  height: 90px;
  width: 100%;
  position: absolute;

  @media (max-width: 980px) {
    height: auto;

    h1 {
      max-width: 50%;
      margin: 0 auto;

      img {
        max-width: 100%;
        margin: .5em auto;
        display: block;
      } // fim do img
    } // fim do h1
  } // fim do media querie
} // fim do header

```

O `header` no arquivo compilado ficará assim:

```

header {
  border-top: 5px solid #c24e4b;
  background: rgba(30, 44, 53), 0.8);
  height: 90px;
  width: 100%;
  position: absolute; }

@media (max-width: 980px) {
  header {
    height: auto; }

  header h1 {
    max-width: 50%;
    margin: 0 auto; }

  header h1 img {

```



```
max-width: 100%;  
margin: .5em auto;  
display: block; } }
```

Assim, o Sass facilitou mais uma vez algo que nem esperávamos que fosse necessário facilitar: a visualização dos *media queries*, permitindo a internalização destes dentro de seus seletores. Aproveitemos essa ajuda.