

Projeto Portal CompJúnior

28 de Fevereiro de 2014

Documento de padronização de código

Comp Júnior

Comp Júnior – Versão 1.0

Harryson de Carvalho Guimarães

(Analista de Requisitos)

harryson.guimaraes@compjunior.com.br

(35) 9955-9984



APRESENTAÇÃO

Este documento tem o objetivo de servir aos desenvolvedores do Portal CompJúnior como um guia a ser seguido para a escrita de código coeso, de forma que toda linha de código pareça ter sido escrita por uma única pessoa, independente da quantidade de desenvolvedores do projeto.

Sumário

1	REGRAS GERAIS
2	HTML
3	CSS
4	CORES
5.....	TIPOGRAFIA

Versão 1.0	Versão inicial – Harryson Guimarães
------------	-------------------------------------

1 REGRAS GERAIS

COMENTÁRIOS

Código é escrito e mantido por pessoas. Certifique-se de que o código é descritivo, bem comentado e amigável para outros.

Grandes pedaços de comentários devem ter contexto e não devem apenas reiterar um nome de classe ou componente.

```
/* Exemplo ruim */  
/* container header */  
.container-header {  
    ...  
}
```

```
/* Exemplo bom */  
/* Elemento para centralizar o header em  
960px */  
.container-header {  
    ...  
}
```

ORGANIZAÇÃO

- Organize seções do código por componentes.
- Desenvolva uma hierarquia de comentários consistente.
- Se usar múltiplos arquivos CSS, quebre-os em componentes.

2 HTML

- Para indentar o código, use tabulação. Não use espaço.
- Elementos aninhados devem estar identados uma vez.
- Sempre use aspas duplas, nunca use aspas simples.
- Nunca inclua uma barra invertida nos elementos viúvos (exemplo: `` ou `<hr>`).

EXEMPLO:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Comp Júnior</title>
  </head>
  <body>
    
    <h1 class="hello-world">Hello,
world!</h1>
  </body>
</html>
```

DOCTYPE DO HTML5

Reforce os padrões com este simples doctype no início de todas as páginas HTML:

```
<!DOCTYPE html>
```

ORDEM DOS ATRIBUTOS

Atributos HTML devem usar esta ordem em particular para facilitar a leitura do código:

- **class**
- **id**
- **data-***
- **for, type, ou href**

EXEMPLO:

```
<a class="..."id="..." data-modal="..."
href="#"></a>
```

```
<input class="form-control" type="text">
```

```

```

REDUZINDO O MARKUP

Sempre que possível, evite elementos pais supérfluos quando escrever HTML. Muitas vezes é necessário uma interação ou refatoração, mas isso produz pouco HTML. Veja o seguinte exemplo:

```
<!-- Não é bom -->
<span class="avatar">
  
</span>
```

```
<!-- Melhor opção -->

```

3 CSS

- Para indentar o código, use tabulação. Não use espaço.
- Quando agrupar seletores, deixe seletores individuais em uma linha.
- Inclua um espaço antes de abrir um colchete.
- Coloque o fechamento do colchete em uma nova linha.
- Inclua um espaço depois dos dois pontos : em cada declaração de propriedade.
- Cada declaração deve estar em sua própria linha para encontrarmos melhor os erros.
- Feche todas as declarações com ponto-vírgula ;.
- Deixe todos os valores hexadecimais em caixa baixa, exemplo: `#fff`.
- Use cores em hexadecimal abreviadas quando puder, exemplo: `#fff` ao invés de `#ffffff`.
- Coloque aspas nos valores em seletores com atributos, exemplo: `input[type="text"]`.
- Evite especificar unidades quando os valores forem 0, exemplo: `margin: 0;` em vez de `margin: 0px;`.
- Separe as classes de CSS com nome composto por “ – “. Exemplo: `.borda-direita{...}`
- Mantenha as classes em caixa baixa e use traços (não underscores ou camelCase).

- Evite usar arbitrariamente notações abreviadas
- Mantenha as classes pequenas e sucintas ao máximo possível.
- Use nomes com significado; tente usar nomes ligados ao visual.

EXEMPLO:

```
/* CSS ruim */
.seletor, .seletor-secundario,
.seletor[type=text] {
  padding:15px;
  margin:0px 0px 15px;
  background-color:rgba(0, 0, 0, 0.5);
  box-shadow:0 1px 2px #CCC,inset 0 1px
0 #FFFFFF
}
```

```
/* CSS bom */
.seletor,
.seletor-secundario,
.seletor[type="text"] {
  padding: 15px;
  margin: 0 0 15px;
  background-color: rgba(0,0,0,.5);
  box-shadow: 0 1px 2px #ccc, inset 0
1px 0 #fff;
}
```

```
/* Exemplo ruim*/
.t { ... }
.red { ... }
```

```
.header { ... }

/* Exemplo bom */
.borda-direita-preta { ... }
```

ORDEM DE DECLARAÇÃO

Declarações relacionadas devem ser agrupadas, colocando posicionamento e as propriedades de box-model perto do topo, seguido das propriedades de tipografia e depois propriedades visuais. EXEMPLO:

```
.ordem-declaracao {
  /* Posicionamento */
  position: absolute;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: 100;

  /* Box-model */
  display: block;
  float: right;
  width: 100px;
  height: 100px;

  /* Tipografia */
  font: normal 13px "Helvetica Neue", sans-serif;
  line-height: 1.5;
  color: #333;
  text-align: center;
```



```

    /* Visual */
    background-color: #f5f5f5;
    border: 1px solid #e5e5e5;
    border-radius: 3px;

    /* Outros */
    opacity: 1;
  }

```

ONDE COLOCAR AS MEDIA QUERIES

Coloque as Media Queries o mais perto possível de suas regras originais. Não as coloque separadas em outros arquivos ou no final do documento. Se você fizer isso, outros poderão não encontrá-las no futuro. Veja um exemplo típico:

```

.elemento { ... }
.elemento-avator { ... }
.elemento-selecionado { ... }

@media (min-width: 480px) {
  .elemento { ... }
  .elemento-avator { ... }
  .elemento-selecionado { ... }
}

```

PROPRIEDADES COM PREFIXOS

Quando usar prefixos de browsers, indente cada propriedade alinhada verticalmente para facilitar a edição multi-linha:

```

.seletor {
  -webkit-box-shadow: 0 1px 2px
    box-shadow: 0 1px 2px
}

```

DECLARAÇÕES ÚNICAS

Em lugares onde são definidas apenas uma linha de propriedade, considere remover as quebras de linha para melhorar a leitura e edição. Considere este exemplo:

```
/* Declarações únicas */
.span1 { width: 60px; }
.span2 { width: 140px; }
.span3 { width: 220px; }

.sprite {
  display: inline-block;
  width: 16px;
  height: 15px;
  background-image: url(../img/sprite.png);
}
.icone      { background-position: 0 0; }
.icone-home { background-position: 0 -20px; }
}
.icone-base { background-position: 0 -40px; }
}
```

4 CORES





5 TIPOGRAFIA

A tipografia utilizada no layout foi Yanone Kaffeesatz e suas variações:

- Extra-light
- Light
- Regular
- Bold