



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

TCC I: QUALIDADE DE CÓDIGO CSS

VICTOR CARNEIRO SALVADOR

Orientador: Prof. Flávio Cruzeiro
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG

BELO HORIZONTE
ABRIL DE 2015

VICTOR CARNEIRO SALVADOR

TCC I: QUALIDADE DE CÓDIGO CSS

BELO HORIZONTE

ABRIL DE 2015

Resumo

Síntese do trabalho em texto cursivo contendo um único parágrafo. O resumo é a apresentação clara, concisa e seletiva do trabalho. No resumo deve-se incluir, preferencialmente, nesta ordem: brevíssima introdução ao assunto do trabalho de pesquisa (qualificando-o quanto à sua natureza), o que será feito no trabalho (objetivos), como ele será desenvolvido (metodologia), quais serão os principais resultados e conclusões esperadas, bem como qual será o seu valor no contexto acadêmico. Para o projeto de dissertação sugere-se que o resumo contenha até 200 palavras.

Palavras-chave: latex. abntex. modelo. (Entre 3 a 6 palavras ou termos, separados por ponto, descritores do trabalho. As palavras-chaves são Utilizadas para indexação.

Abstract

Translation of the abstract into english, possibly adapting or slightly changing the text in order to adjust it to the grammar of english educated.

Keywords: latex. abntex. template.

Sumário

1 – Introdução	1
1.1 Objetivos	2
2 – Trabalhos Relacionados	3
3 – HTML	4
3.1 Revendo os conceitos	4
4 – CSS	5
4.1 Revendo os conceitos	5
4.1.1 Seletores	5
4.1.2 Efeito Cascata	6
5 – Metodologia	7
5.1 Questionário	7
5.2 Proposta das Métricas	7
5.3 Avaliação dos resultados	8
6 – Resultados Esperados	9
7 – Conclusão	10
Referências	11

1 Introdução

Linguagens de folha de estilo são muito pouco documentadas ([MARDEN; MUNSON, 1999](#)). Este fato continua sendo verdadeiro, como identificado nos trabalhos de [Geneves et al. \(2012\)](#) e [Quint e Vatton \(2007\)](#). Pode-se apontar algumas características do *Cascading Style Sheet* (CSS) — formato de apresentação de estilo, padronizado pela *World Wide Web Consortium* (W3C) — como sendo complicadores para autoria de estilos CSS.

Com o aumento da popularidade de páginas *web*, essa tecnologia passou a ser utilizada para criação de páginas mais complexas do que simples documentos, *e.g.*, portais de venda, fóruns, portais de vídeos, entre outras. A autoria de documentos CSS, muitas vezes, é referida como codificação, já que, na maioria das vezes, editar o código CSS é mais custoso que a criação do próprio desenho.

Sendo um dos três padrões fundamentais da W3C para desenvolvimento de conteúdo *web*, o CSS se tornou largamente utilizado. A separação do documento de conteúdo da apresentação, foi certamente o principal motivo do CSS se tornar tão popular. Apesar das vantagens dessa separação estrutural, os códigos se tornaram complexos e de manutenibilidade onerosa ([MESBAH; MIRSHOKRAIE, 2012](#)).

Escrever regras CSS não é uma tarefa trivial, as características da linguagem como herança e especificidade de seletores, colocam os desenvolvedores constantemente em situações nas quais se questionam a efetividade das associações de propriedades escolhidas. Essas características podem prejudicar o que [Keller e Nussbaumer \(2010, p. 116\)](#) definem como efetividade e eficiência:

- **Efetividade do código:** a folha de estilo é efetiva se o documento de conteúdo ao qual ele é aplicado renderiza da forma desejada.
- **Eficiência do código:** folhas de estilo que causam o mesmo efeito em um documento de conteúdo ainda pode diferir significativamente no modo em que ela aplica a associação de propriedade. Maximizar a eficiência do código CSS significa aplicar a associação de propriedades de uma forma que o esforço da autoria, manutenção e eventual reutilização seja minimizado.

Pode-se notar a dificuldade de se manter um código CSS sem falhas durante a construção de uma página *web*, exigindo-se um alto grau de manutenibilidade. A manutenibilidade de um sistema é a facilidade com a qual um sistema de *software*, ou componente, pode ser modificado para corrigir falhas, melhorar performance, ou adaptar-se à mudança de ambiente ([IEEE, 1990](#)). A partir desta definição, pode-se identificar uma medida de manutenibilidade para códigos

CSS, considerando-se que onde houver alta complexidade haverá a necessidade de se manter o funcionamento, ou adaptação, da apresentação do documento.

Como identificado por [Mesbah e Mirshokraie \(2012\)](#), analisar código CSS com uma perspectiva de manutenção ainda não foi explorada em nenhum trabalho científico. Portanto, há necessidade de se definir a qualidade do código CSS, com objetivo de se manter um nível de manutenibilidade da apresentação de páginas *web*. A proposta apresentada se refere à uma medição da facilidade de manutenção da folha de estilo.

1.1 Objetivos

A manutenção e modificação de *software* são etapas essenciais para o seu tempo de vida, e isso não é diferente para aplicações *web*. Sendo uma tarefa essencial, e complexa, entende-se que seja necessário encontrar uma forma de mitigar os possíveis impactos na modificação, ou evolução, das folhas de estilo dos projetos *web*.

Pretende-se encontrar uma forma de se medir, de forma quantitativa, a manutenibilidade de um código CSS. Para tanto, será feito um levantamento das propriedades da linguagem que modificam a facilidade da manutenção de seu produto, levando em conta tanto a visão do autor, quanto dos aspectos funcionais.

Este trabalho tem por objetivo encontrar métricas de qualidade de código fonte CSS, com o intuito de melhorar a manutenibilidade do código, garantindo as propriedades necessárias para tal, como legibilidade e entendimento, evitando ainda os possíveis efeitos colaterais que qualquer modificação possa causar.

2 Trabalhos Relacionados

Existem poucos trabalhos que tratem de qualidade de código CSS na literatura, e como [Mesbah e Mirshokraie \(2012\)](#) identificou, não existem trabalhos que analisem o código em função da manutenibilidade.

Existem trabalhos como o de [Keller e Nussbaumer \(2010\)](#), que analisa a qualidade de código CSS em uma perspectiva de avaliar a diferença em códigos de autoria humana e os auto gerados. Enquanto [Mesbah e Mirshokraie \(2012\)](#) propõem uma ferramenta para auxiliar na manutenção de código encontrando regras inefetivas e removendo-as do código.

[Keller e Nussbaumer \(2010\)](#) propõem uma medida de qualidade do código CSS baseando-se na abstração do seletor. Esse trabalho é baseado no argumento de que o objetivo do código CSS é a reutilização de suas regras, a abstração do seletor é então definida pela sua utilização no escopo geral do HTML, considerando que seletores com id são os menos abstratos possíveis. Este trabalho não conseguiu encontrar uma relação forte com a complexidade de código CSS e o nível de abstração, de forma que os autores a consideraram uma medida fraca, se utilizada de forma exclusiva, deixando em aberto a proposta de métricas que a corroborem, ou cooperem com ela na medida de qualidade do código CSS.

Em [Quint e Vatton \(2007\)](#) ferramentas de auxílio na autoria de CSS são avaliadas, e a forma de avaliação do código, como estrutura de dados e método de avaliação do código. Esse trabalho propõe uma ferramenta didática, que auxiliará na criação de folhas de estilo, auxiliando o entendimento da aplicação das propriedades e regras. Neste trabalho são também identificados fatores que tornam a tarefa de autoria de código CSS não trivial.

3 HTML

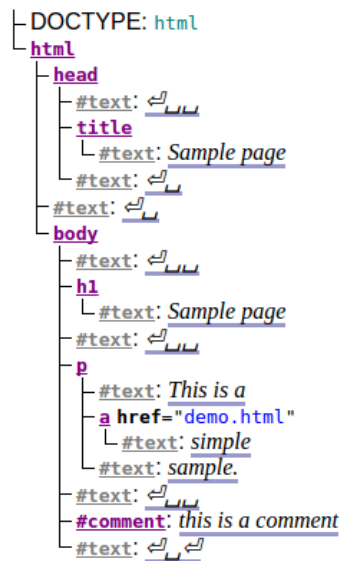
3.1 Revendo os conceitos

HTML é a linguagem principal para criação de documentos, e aplicações, na *Web*, para o uso de todos, em qualquer lugar (W3C, 2015b).

O documento HTML consiste em uma árvore de elementos e texto. Cada elemento é representado por uma *tag* de abertura e uma de fechamento. As *tags* têm de estar todas aninhadas completamente, sem haver sobreposição. Os elementos podem ter atributos que controlam o comportamento do elemento (HICKSON et al., 2014).

Os navegadores *web* (*Browsers*) traduzem esse formato em uma árvore DOM (*Document Object Model*). Uma árvore DOM é uma representação em memória de um documento, que possui vários nós, cada nó agrupando alguns atributos do documento.

Figura 1 – Exemplo de estrutura da árvore do DOM



Fonte: Hickson et al. (2014)

Na figura Figura 1 vê-se que o elemento raiz da árvore é o html, já que este é o tipo do documento interpretado. Cada elemento do html é representado por um nó, todos os elementos que se encontram no nível abaixo deste são denominados descendentes (*descendant*), os nós que se encontram exatamente um nível abaixo são os filhos (*child*), e os elementos que se encontram no mesmo nível são chamados de irmãos (*sinbling*). Os nós demonidados de text são os nós que encapsulam os textos inseridos dentro dos elementos html, então estes serão os nós que existiram em maior número no DOM.

4 CSS

4.1 Revendo os conceitos

CSS é um mecanismo simples para adicionar estilo (*e.g.*, fontes, cores, espaçamento) em documentos *Web* (W3C, 2015a).

Uma folha de estilo C pode ser vista como um conjunto de regras R , composto por regras simples R_i , cada uma composta por um seletor S_i e um conjunto de pares: propriedade P_i e seus valores V_i . Os seletores definem a quais elementos de um documento, serão aplicadas as propriedades definidas pela regra à qual elas pertencem (GENEVES et al., 2012).

4.1.1 Seletores

Um seletor é uma cadeia de uma, ou mais, sequências de seletores simples, separados por combinadores. Os seletores simples são cadeias de caracteres que representam um elemento do html: o seletor universal, representado pelo símbolo $*$, indica que a regra será aplicada a todos os elementos que estejam no DOM. O seletor de elementos html, representado pelo nome da *tag* de um elemento, por exemplo $h1$. O seletor de classe, que seleciona todos os elementos que possuam o atributo *class* especificado pelo seletor, é utilizado escrevendo-se o nome da classe, precedido pelo símbolo $\#$. O seletor de *id*, seleciona o elemento do html que possua aquele *id*, é utilizado escrevendo-se o identificador precedido de um ponto final ($.$). Simplificando, sem perder generalidade, podemos considerar que regras são feitas de seletores únicos que definem uma única propriedade por vez. Os seletores S_i , chamados de padrões na especificação do CSS (ÇELIK et al., 2009), definem uma função booleana na forma:

$$expression * element \rightarrow boolean \quad (1)$$

que define se um elemento é, ou não, selecionado pela expressão do seletor.

Os combinadores são propriedades que definem relações entre os elementos de um documento. Existem três formas de combinadores: descendentes, filhos e irmãos. O combinador de descendente descreve qualquer elemento que esteja um nível abaixo na árvore DOM, e são representados pelo espaço em branco, *e.g.* "*body p*". O combinador de filho descreve os elementos que estão exatamente um nível abaixo do nó, este combinador é representado pelo sinal de maior ($>$), *e.g.* "*body > p*". O combinador de irmãos descreve os elementos que estão no mesmo nível da árvore, existem duas variações, uma para o próximo irmão adjacente ($+$) e um para todos os irmãos (\sim).

4.1.2 Efeito Cascata

O efeito cascata do CSS se dá devido à ordem de precedência dos valores de propriedades definidas para cada elemento. O mecanismo recebe uma lista desordenada destas propriedades, e as organiza pela precedência das declarações delas. Essa ordem é definida de acordo com os critérios listados abaixo, em ordem decrescente de prioridade (FANTASAI; ATIKNS, 2015).

- **Origem e Importância:** Cada regra de estilo possui uma origem, que define onde ela estará na cascata, e a importância se refere à utilização, ou não, do atributo *!important*.
- **Escopo:** Uma declaração pode ter uma subárvore do DOM como escopo, afetando somente os elementos pertencentes a esta subárvore. Para declarações normais, o escopo mais interno tem prioridade, para regras com *!important* as regras do escopo mais externo sobrescreverá.
- **Especificidade:** O calculo de especificidade conta a ocorrência de seletores de ID, classe e tipo (*tags* e *pseudo-elements*), e faz-se uma soma ponderada dessas ocorrências. A declaração com maior especificidade tem prioridade.
- **Ordem de aparição:** A última declaração no documento tem a maior prioridade. Isto significa que a localidade será levada em conta, para isso, considera-se que as folhas de estilo são concatenadas ao documento na ordem em que são declaradas.

5 Metodologia

O primeiro passo da execução do trabalho consistirá na consolidação do conceito de manutenibilidade em código CSS. Pretendemos alcançar este objetivo por meio de referências bibliográficas e de uma pesquisa do tipo *survey* com desenvolvedores profissionais, com níveis de experiência variados. Esta pesquisa pretende identificar as propriedades da linguagem e as situações mais comuns que dificultam, ou facilitam, a manutenção de códigos CSS.

Após executada a pesquisa, as métricas serão consolidadas, propondo valores individuais para as características da linguagem e definindo um índice de qualidade a partir de um agrupamento destes valores. Dessa forma serão obtidas informações suficientes para construir uma ferramenta para validar as métricas definidas.

Utilizando de ferramentas automatizadas, serão identificadas as relações entre índice proposto e a quantidade de falhas identificadas em sistemas *web* de código aberto. Analisando estes projetos, iremos fazer a referência cruzada, entre a pontuação alcançada pelo código CSS e o número de problemas reportados relacionados ao mesmo.

5.1 Questionário

A pesquisa *survey* (questionário) é uma forma de obtenção de dados ou informações sobre características, ações ou opiniões de um determinado conjunto de pessoas, indicado como representante de uma população-alvo, por meio de um instrumento de pesquisa, normalmente um questionário (FREITAS; OLIVEIRA, 2000).

O interesse de uma pesquisa desse tipo é produzir descrições quantitativas de uma população. No caso deste trabalho, o objetivo é identificar, de forma quantitativa, as características identificadas pelos desenvolvedores como sendo as que classificam a legibilidade do código. Para tanto, será executado um questionário exploratório, com o objetivo de identificar os conceitos do CSS que são centrais para a associação de qualidade do código.

5.2 Proposta das Métricas

As métricas são identificadores numéricos baseados em características da linguagem. No caso do CSS, estas características serão definidas a partir dos resultados obtidos no questionário.

5.3 Avaliação dos resultados

A partir da métrica proposta será desenvolvida uma ferramenta de validação automática. O programa irá ler o arquivo CSS, identificará as regras definidas e, a partir da definição proposta, será calculado para cada arquivo uma métrica. Se necessário, os arquivos HTML também serão considerados no cálculo da métrica.

Com as métricas calculadas, será testada a aderência da métrica a projetos *open source*, utilizando o número de problemas relacionados a CSS como parâmetro de triangulação. Dessa forma será construída uma base de dados para as análises estatísticas, que confrontaram os resultados esperados deste trabalho.

6 Resultados Esperados

Entende-se que códigos inutilizados são o limite inferior da escala, uma vez que eles só ocuparam espaço no documento, dificultando a leitura e identificação de efeitos colaterais, se houverem. Por sua vez, propriedades que não sobrescrevam nenhuma outra, são o limite superior da escala. Muitas características da linguagem podem reduzir o valor dessas últimas, como especificidade do seletor, profundidade, localidade, etc.

Espera-se poder identificar de forma empírica quais são qualidades de manutenibilidade que atendam ao objetivo, de forma que nossos índices possam auxiliar na escrita de códigos que sejam modificados sem causar problemas adicionais.

Espera-se que a utilização das métricas propostas não influenciem de forma negativa em questões de desempenho, apesar de esta não ser uma característica vital da pesquisa aqui projetada.

Uma métrica válida de manutenção seria inédita na literatura, como identificado por [Mesbah e Mirshokraie \(2012\)](#). Este trabalho pretende encontrar uma norma que identifique a manutenibilidade das folhas de estilo, que será uma grande contribuição para a comunidade de desenvolvedores *web* e *designers*.

7 Conclusão

Referências

- FANTASAI; ATIKNS, T. **CSS Cascading and Inheritance Level 4**. 2015. Disponível em: <http://www.w3.org/TR/2015/WD-css-cascade-4-20150421/>. Acesso em: 27 de maio de 2015. Citado na página 6.
- FREITAS, H.; OLIVEIRA, M. **O Método de pesquisa Survey**. 2000. 105–112 p. Disponível em: <http://www.rausp.usp.br/download.asp?file=3503105.pdf>. Citado na página 7.
- GENEVES, P.; LAYAIDA, N.; QUINT, V. On the analysis of cascading style sheets. In: **Proceedings of the 21st international conference on World Wide Web - WWW '12**. New York, New York, USA: ACM Press, 2012. p. 809. ISBN 9781450312295. Disponível em: <http://dl.acm.org/citation.cfm?id=2187836.2187946>. Citado 2 vezes nas páginas 1 e 5.
- HICKSON, I.; BERJON, R.; FAULKNER, S.; LEITHEAD, T.; NAVARA, E. D.; O'CONNOR, E.; PFEIFFER, S. **HTML5: A vocabulary and associated APIs for HTML and XHTML**. 2014. Disponível em: <http://www.w3.org/TR/html/introduction.html#a-quick-introduction-to-html>. Acesso em: 24 de maio de 2015. Citado na página 4.
- IEEE. **IEEE Standard Glossary of Software Engineering Terminology**. 1990. 1 p. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342. Citado na página 1.
- KELLER, M.; NUSSBAUMER, M. Css code quality: A metric for abstractness or why humans beat machines in css coding. In: **Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the**. [S.l.]: IEEE, 2010. p. 116–121. ISBN 978-1-4244-8539-0. Citado 2 vezes nas páginas 1 e 3.
- MARDEN, P.; MUNSON, E. Today's style sheet standards: the great vision blinded. **Computer**, v. 32, n. 11, p. 123–125, 1999. ISSN 00189162. Disponível em: http://www.researchgate.net/publication/2955166_Today's_style_sheet_standards_the_great_vision_blinded. Citado na página 1.
- MESBAH, A.; MIRSHOKRAIE, S. Automated analysis of css rules to support style maintenance. In: **Proceedings of the 34th International Conference on Software Engineering**. Piscataway, NJ, USA: IEEE Press, 2012. (ICSE '12), p. 408–418. ISBN 978-1-4673-1067-3. Disponível em: <http://dl.acm.org/citation.cfm?id=2337223.2337272>. Citado 4 vezes nas páginas 1, 2, 3 e 9.
- QUINT, V.; VATTON, I. Editing with style. In: **Proceedings of the 2007 ACM symposium on Document engineering - DocEng '07**. New York, New York, USA: ACM Press, 2007. p. 151. ISBN 9781595937766. Disponível em: <http://dl.acm.org/citation.cfm?id=1284420.1284460>. Citado 2 vezes nas páginas 1 e 3.
- W3C. **CSS**. 2015. Disponível em: <http://www.w3.org/Style/CSS>. Acesso em: 21 de maio de 2015. Citado na página 5.
- W3C. **HTML**. 2015. Disponível em: <http://www.w3.org/html/>. Acesso em: 24 de maio de 2015. Citado na página 4.
- ÇELIK, T.; ETEMAD, E. J.; GLAZMAN, D.; HICKSON, I.; LINSS, P.; WILLIAMS, J. **Selectors Level 3**. 2009. Disponível em: <http://www.w3.org/TR/2009/PR-css3-selectors-20091215/>. Acesso em: 24 de maio de 2015. Citado na página 5.