

UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

GUILHERME CORRÊA MORGADO

**SASS4J: UM COMPILADOR SASS
PARA A PLATAFORMA JAVA EE**

Rio Grande - RS

2013

GUILHERME CORRÊA MORGADO

**SASS4J: UM COMPILADOR SASS
PARA A PLATAFORMA JAVA EE**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para obtenção do grau
de Tecnólogo em Análise e Desenvolvimento
de Sistemas.

Profº. Márcio Josué Ramos Torres
Orientador

Rio Grande - RS

2013

FOLHA DE APROVAÇÃO

Monografia sob o título "SASS4J: UM COMPILADOR SASS PARA A PLATAFORMA JAVA EE" defendida por GUILHERME CORRÊA MORGADO e aprovada em XX de XXXXXXXX de 2013, em Rio Grande, estado do Rio Grande do Sul, pela banca examinadora constituída pelos professores:

Profº. Márcio Josué Ramos Torres

Orientador

Profº. Ramiro Saldaña Garin

IFRS - *Campus* Rio Grande

Profº. Ms. Leonardo Vianna do Nascimento

IFRS - *Campus* Rio Grande

RESUMO

A realidade do desenvolvimento *front-end* hoje nas empresas é que há pouco tempo para realização de muitas tarefas. O simples fato de reduzir tempo e aumentar produtividade com uma ferramenta de fácil utilização torna o dia a dia do desenvolvedor mais tranquilo, consequentemente, mais produtivo. Encontrar algo que faça o desenvolvedor repensar na forma de desenvolvimento é uma tarefa árdua e ao mesmo tempo prazerosa, porém é algo que motiva e encorpa o projeto.

Todavia, introduzir este projeto no cotidiano de empresas e principalmente de usuários, necessita na facilidade de uso e na qualidade do código final gerado. Estes obstáculos fazem do projeto uma meta pessoal, tornar o passar dos dias do desenvolvedor em uma tarefa intuitiva e de bom agrado na execução.

Este projeto tem como finalidade a criação de uma saída de um pré-processador bem visto e utilizado nos dias atuais para uma linguagem bem conceituada de mercado, facilitando o dia a dia de um determinado nicho de programadores que utilizam Java EE nos seus projetos.

Palavras-chave:

compilador, sass, servlet, front-end, java ee.

ABSTRACT

The reality of front-end development in companies today is that there is little time for many tasks. The simple fact reduce time and increase productivity with a user-friendly tool makes the everyday developer's quiet and clear, more productive. Find something that makes developer reconsider the form of development is an arduous task and while enjoyable, but it is something that motivates and embodies the design.

However, introducing this design in everyday business and especially users need ease of use and quality of the final code generated. These obstacles make the project a personal goal, make the days go by the developer on a task and intuitive gladly execution.

This project aims to create an output of a preprocessor well regarded and utilized today for a reputable language market, facilitating the daily life of a particular niche developers using Java EE in their projects.

Keywords:

compiler, sass, servlet, front-end, java ee.

LISTA DE FIGURAS

FIGURA 1 - Exemplo de código HTML com classes CSS	13
FIGURA 2 - Exemplo de código CSS gerado pelo Sass	14
FIGURA 3 - Exemplo de código escrito em Sass e o código gerado a partir deste	14
FIGURA 4 - Exemplo de código fonte utilizando Java EE.....	16
FIGURA 5 - Estrutura de criação de classes do padrão Composite.....	18

LISTA DE SIGLAS

ANTLR	ANother Tool for Language Recognition
API	Application Programming Interface
CSS	Cascading Style Sheets
DSL	Domain Specific Language
HTML	HyperText Markup Language
Sass	Syntactically Awesome Stylesheets
WWW	World Wide Web
W3C	World Wide Web Consortium

SUMÁRIO

1 - Introdução.....	10
1.1 - O que é o projeto	10
1.2 - Para que serve.....	11
1.3 - Público alvo	11
1.4 - Objetivos.....	11
1.4.1 - Objetivos específicos	11
1.5 - Partes em que se divide	11
1.6 - Resultados esperados.....	12
1.7 - Embasamento teórico	12
1.7.1 - Web	12
1.7.2 - HTML	12
1.7.3 - CSS.....	13
1.7.4 - Sass.....	14
1.7.5 - API	15
1.7.6 - DSL	15
1.7.7 - Servlet	15
1.7.8 - Java EE.....	15
1.7.9 - Compiladores	16
1.8 - Padrões de projetos.....	16
1.8.1 - Interpreter.....	17
1.8.2 - Composite	17
1.9 - Tecnologias Relacionadas.....	18
1.9.1 - LESS	18
1.9.2 - Stylus.....	18
1.9.3 - Jsass.....	18
2 - Referências	19

1. INTRODUÇÃO

Quando o desenvolvedor está escrevendo um código *front-end* para um novo projeto, há a repetição de algumas ações básicas desse cenário, como a criação de novas classes de estilos. Este tempo gasto com repetições se torna alto em relação ao tamanho do projeto, ou seja, quanto maior a necessidade de criação de páginas, maior será o tempo gasto com repetições no código. Com o intuito de reduzir este tempo gasto com repetições, foram criadas no passado recente algumas ferramentas de auxílio, mais conhecidos como preprocessadores de Cascading Style Sheets (CSS). Esses preprocessadores recebem instruções para facilitar a repetição de códigos, a edição de um código ou até o reaproveitamento de parte de um código. Sendo assim, podendo ser aproveitado um trecho de código quantas vezes forem necessárias, a operação de edição de um código se torna mais fácil e até o reaproveitamento de parte de um código para ser utilizado em outro módulo se torna uma operação menos sucessível a erros e falhas.

1.1. O QUE É O PROJETO

O projeto visa ajudar o desenvolvedor que utiliza estes preprocessadores, visto que estas ferramentas não possuem saídas existentes em todas as linguagens de programações existentes. O projeto se especifica ao criar um compilador para a plataforma Java EE de um preprocessador já escolhido: o Syntactically Awesome Stylesheets (Sass). Para com isso colaborar com a produtividade dos desenvolvedores que utilizem a combinação desta ferramenta com a linguagem escolhida.

A motivação para realizar este projeto partiu no momento em que observou-se o tempo que era gasto com operações simples de edição de códigos CSS. Percebeu-se que o tempo gasto copiando e colando códigos que já estavam prontos, estava excessivamente longo. Assim, pensou-se sobre a possibilidade de haver otimização do tempo. Ao ser orientado, foi possível integrar a necessidade de um desenvolvedor *front-end*, com a lacuna que há entre este preprocessador e a plataforma de programação utilizada.

1.2. PARA QUE SERVE

Um preprocessador de CSS é uma forma de desenvolver CSS de forma mais produtiva. O Sass além de fornecer uma forma mais simples e produtiva de escrever CSS, possibilita a criação de arquivos separados, organizados, que após serem compilados, formam um único arquivo CSS. Além de organização e produção, Sass garante a criação de funções, uso de variáveis e expressões matemáticas. Hoje, o Sass é compilado somente via Ruby. A utilidade deste projeto é criar uma versão do compilador nativa em Java EE, para que aplicações em Java para Web possa utilizar esse preprocessador e haja um ganho em tempo de produção.

1.3. PÚBLICO ALVO

O público alvo é toda a comunidade desenvolvedora que utiliza o Sass como seu preprocessador de CSS e utiliza a plataforma Java EE como plataforma de desenvolvimento.

1.4. OBJETIVOS

O objetivo geral do projeto é basicamente analisar e desenvolver um compilador utilizando Java EE para o já existente preprocessador de CSS chamado Sass.

1.4.1. OBJETIVOS ESPECÍFICOS

- Aumentar a produtividade do desenvolvedor de *front-end*;
- Criar uma nova saída para utilização do Sass em uma linguagem de programação sem suporte ao Sass hoje;
- Difundir a utilização de outras ferramentas como frameworks e preprocessadores para a comunidade em geral.

1.5. PARTES EM QUE SE DIVIDE

O projeto divide-se em algumas simples etapas, são elas: definição do escopo do projeto, elaboração do cronograma, execução da descrição textual do projeto, a execução da implementação do projeto e os testes integrados do sistema.

1.6. RESULTADOS ESPERADOS

Os resultados esperados são o completo funcionamento do compilador em uma aplicação feita em Java EE e a criação de uma forma de integrar o preprocessador ao dia a dia do desenvolvedor *back* e *front-end*.

1.7. EMBASAMENTO TEÓRICO

Para maior verossimilhança do projeto, foram selecionados conteúdos que auxiliam o desenvolvimento deste projeto. Seguem conteúdos sobre a arquitetura de desenvolvimento, as linguagens utilizadas e tecnologias relacionadas.

1.7.1. WEB

Teia mundial, tradução de World Wide Web, são basicamente documentos que são manipulados e executados na Internet. Para a visualização destes documentos, são utilizados navegadores, que por si descarregam essas informações de servidores para mostrar o resultado ao usuário. A Web segue o preceito de navegação, o qual diz que o usuário tem a possibilidade de seguir um fluxo de páginas por meio de interligações entre as mesmas.

Na prática, a Web é o ambiente em que poderá ser possível visualizar a proposta desse projeto, ou seja, uma página aberta em um navegador com estilos feitos a partir de um preprocessador utilizando um compilador Java EE para interpretar o código gerado pelo processador.

1.7.2. HTML

HyperText Markup Language (HTML) é a linguagem de marcação mais utilizada para criação de páginas Web. O HTML é mundialmente conhecido por sua compatibilidade com navegadores de internet. O documento é escrito usando *tags* pré-definidas pela linguagem, e suas especificações são escritas atualmente pelo World Wide Web Consortium (W3C), órgão que padroniza a Web.

A finalidade de usar HTML neste projeto é para ser possível estilizar elementos utilizando o Sass, ou seja, o código que será gerado via Sass e compilado utilizando a plataforma Java EE, será utilizado estilizando um conjunto de *tags* HTML. Para estilizar *tags* HTML, são usadas classes CSS, como mostra a **FIGURA 1**.

FIGURA 1 - Exemplo de código HTML com classes CSS.

```

<section id="canais-atendimento">
  <div class="wrapper">
    <div class="col-canais first">
      <h6 title="Canais De Atendimento">Canais De Atendimento:</h6>
    </div>
    <div class="col-canais chatonline">
      <a class="ico ico-chat-online" href="javascript:void(0);" title="Chat On-line">Chat On-line</a>
    </div>
    <div class="col-canais">
      <a class="ico ico-confira-telefones" href="javascript:void(0);" title="Confira Os Telefones">Confira os Telefones</a>
    </div>
    <div class="col-canais">
      <a class="ico ico-envie-mensagem" href="javascript:void(0);" title="Envie Uma Mensagem">Envie uma Mensagem</a>
    </div>
    <div class="col-canais last">
      <a class="ico ico-unidades-venda" href="javascript:void(0);" title="Unidades De Venda">Unidades de Venda</a>
    </div>
    <br class="clr" />
  </div>
</section>

```

1.7.3. CSS

Cascading Style Sheets é utilizado como forma de apresentação de documentos escritos com linguagens de marcação, fornecendo estilos para melhor apresentação das informações.

A principal vantagem de se utilizar CSS é o fato de manter separado a camada de apresentação do conteúdo do documento. O CSS é um formato abrangido por todos os maiores navegadores de mercado, tornando seu uso presente nos dias atuais.

Hoje em dia, o CSS é regulamentado pela W3C, consórcio que monitora, controla e padroniza a Web. O principal objetivo do consórcio é desenvolver protocolos e diretrizes que garantam seu crescimento de longo prazo.

CSS será o código final gerado pelo Sass após sua compilação em uma saída Java EE. O motivo de estilizarmos os textos de marcação utilizando CSS foi devido a saída de código do pré-processador. Para ter mais ideia sobre um código CSS, pode-se ver um exemplo na **FIGURA 2**.

FIGURA 2 - Exemplo de código CSS gerado pelo Sass.

```

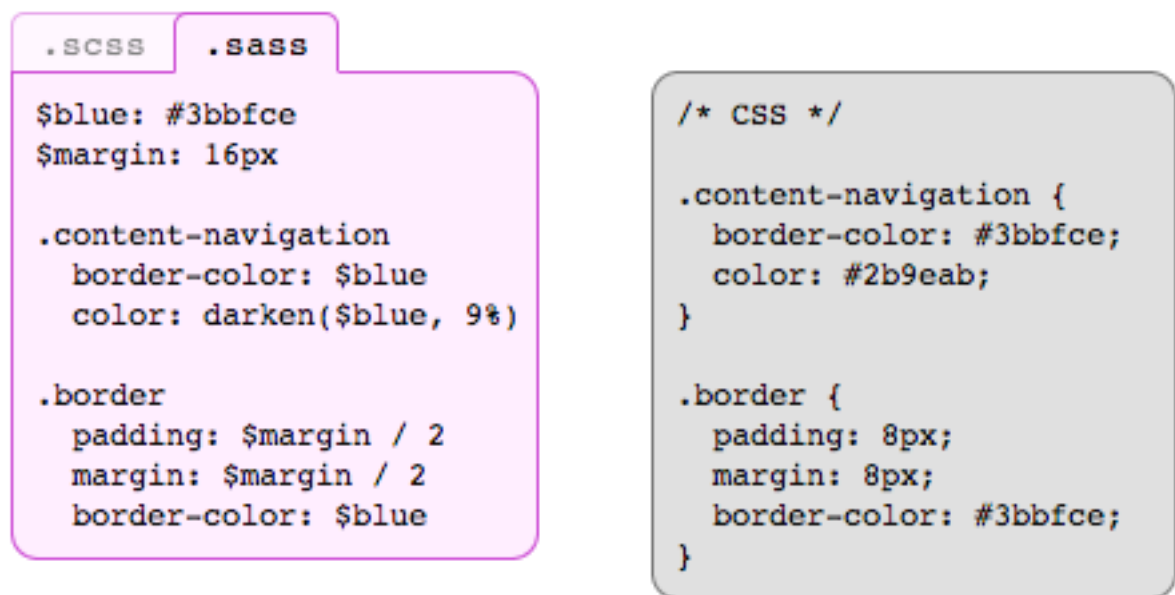
1  .menu {
2      width:100%;
3      border:1px solid #FFF;
4      float:left;
5      border-radius:5px 4px;
6  }
7
8  .menu:hover{
9      font:12px Tahoma;
10     color:#EDEDED;
11 }

```

1.7.4. SASS

O Syntactically Awesome Style Sheets, do inglês, folhas de estilo formidavelmente sintáticas, é uma metalinguagem em cima de CSS que permite escrever folhas de estilos mais limpas e estruturadas. Sass permite a criação de uma folha de estilo mais administrável e com uma sintaxe mais elegante para o CSS, como mostra a **FIGURA 3**.

FIGURA 3 - Exemplo de código escrito em Sass e o código gerado a partir deste.



O Sass é uma linguagem de script que após compilado, gera arquivos em formato CSS. Ele consiste basicamente em duas sintaxes: a original, que é escrita especificamente para o Sass interpretar, e a de saída, que é o próprio CSS. A linguagem interpretada (.sass) têm como características a utilização de elementos para reaproveitamento de código e uso de variáveis. O resultado, também considerado o código CSS final, será gerado sintaticamente mais elegante e de mais fácil compreensão e manuseio.

A utilidade de usar um preprocessador de CSS é o fato de gerar um código completo utilizando um esforço menor. Hoje, o CSS não permite o reaproveitamento inteligente de trechos de código e também não há a possibilidade de utilização de variáveis. Esses pequenos desvios que ocorrem no dia a dia de um desenvolvedor podem ser suprimidos utilizando um preprocessador de CSS.

O Sass também permite que um leque grande de editores de textos possibilitem suporte. Alguns editores renomados estão oferecendo suporte pois veem no Sass como uma alternativa

benéfica ao desenvolvedor na etapa de utilização de folhas de estilo.

1.7.5. API

Uma API é basicamente um conjunto de funções que facilitam o reaproveitamento de código em um sistema ou aplicativo. A API fornece para o usuário um conjunto de regulamentações de como utilizar os métodos que esta aplicação prove, tornando a tarefa do usuário somente estender métodos fornecidos pela própria aplicação.

1.7.6. DSL

Domain Specific Language, inglês de linguagem para domínio específico, é uma linguagem de programação ou de especificação dedicada a um problema particular dentro de um domínio, ou seja, um determinado problema que necessita de uma linguagem específica para solução. Tornar uma DSL útil se dá por meio de criação de uma API (extensa e de fácil utilização, pois essa linguagem específica pode se tornar útil para algum outro usuário ou funcionalidade.

Este projeto tem em vista criar uma DSL para compilar o Sass e gerar um arquivo CSS, tudo isso utilizando a plataforma Java EE.

1.7.7. SERVLET

Para ser possível a utilização de métodos de uma API, será utilizado Servlets disponibilizados pela aplicação. Servlets são classes Java que são usadas para estender as funcionalidades de um servidor ou uma API, além disso, podem responder a quaisquer tipos de requisições, mas a utilização no projeto terá foco no contato entre o compilador a ser desenvolvido e o preprocessor de CSS utilizado.

1.7.8. JAVA EE

Java Platform Enterprise Edition, é uma plataforma de programação para servidores utilizando a linguagem de programação Java. Java EE fornece uma API e um ambiente para o desenvolvimento de softwares multicamadas. A utilização de Java EE neste projeto se dá a necessidade de criar um compilador para Sass em Java EE. Para criarmos essa saída, códigos

fontes serão gerados, utilizando a API desta plataforma, e obedecendo os servlets que serão criados no software. Java EE utiliza como base a linguagem de programação Java, a qual possui uma implementação de exemplo na **FIGURA 4**.

FIGURA 4 - Exemplo de código fonte utilizando Java EE.

```

236 public String showStories() {
237     setCurrentSprint(sprints.getRowData());
238     return "showStories";
239 }
240
241 public String showDashboard() {
242     setCurrentSprint(sprints.getRowData());
243     return "showDashboard";
244 }
245
246 public Sprint getCurrentSprint() {
247     return currentSprint;
248 }

```

1.7.9. COMPILADORES

Compiladores são softwares projetados para lerem códigos fontes escritos em uma linguagem de programação interpretada e transformar em código fonte executável. O compilador será de suma importância neste projeto, visto que será necessário criar um compilador para ler o código gerado em Sass usando Java EE, e retornado um arquivo executável CSS.

1.8. PADRÕES DE PROJETO

Para a melhor apresentação e organização deste projeto, foram escolhidos alguns padrões de projetos que poderão ser adicionados ao projeto no decorrer do desenvolvimento. Serão descritos eles abaixo para saber qual a possível aplicação e utilização de cada.

1.9. TECNOLOGIAS RELACIONADAS

Além de Sass, existem outras ferramentas no mercado para interpretação de CSS. Essas ferramentas normalmente divergem entre si no quesito sintaxe, porém a divergência não é grande.

Ao ser falado da variedade de interpretadores que existem no mercado, também devem ser mencionados os compiladores e interpretadores feitos para a utilização de Sass na plataforma Java EE.

1.9.1. LESS

LESS é uma linguagem de folha de estilos dinâmica. Assim como o Sass, o LESS possui sua sintaxe própria, que após ser interpretada, gera um arquivo CSS de acordo com o que foi escrito. Uma característica marcante sobre o LESS é a possibilidade de rodar o interpretador tanto na aplicação do cliente quanto no servidor.

1.9.2. STYLUS

Stylus é uma ferramenta de préprocessamento de simples instalação e manuseio para gerar CSS. Possui as mesmas características dos outros préprocessadores, o qual processa um arquivo de sintaxe própria para geração de um CSS.

1.9.3. JSASS

Jsass é um compilador de Sass utilizando Java. É baseado na utilização de ANTLR(ANother Tool for Language Recognition) para a interpretação de expressões em Sass. Apesar das poucas versões de códigos enviados, pode-se perceber que é um código de compilação simples, que utiliza de *tokens* para o processo de criação de códigos CSS.

REFERÊNCIAS

GAMMA, E.; HELM, R.; JOHSON, R.; VUSSIDES, J. **Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2000.

KERIEVSKY, J. **Refatoração para Padrões**. Porto Alegre: Bookman, 2008.

Disponível em: <<http://www.tidbits.com.br/desenvolvendo-css-de-forma-mais-produtiva-usando-sass>>. Acesso em: 17 jul. 2013.

Disponível em: <[http://en.wikipedia.org/wiki/Sass_\(stylesheet_language\)](http://en.wikipedia.org/wiki/Sass_(stylesheet_language))>. Acesso em: 17 jul. 2013.

Disponível em: <<https://drupal.org/project/sass>>. Acesso em: 17 jul. 2013.

Disponível em: <<https://vaadin.com/api/7.0.7/com/vaadin/sass/SassCompiler.html>>. Acesso em: 17 jul. 2013.

Disponível em: <<http://sass-lang.com/editors.html>>. Acesso em: 17 jul. 2013.

Disponível em: <<http://tableless.com.br/sass-um-outro-metodo-de-escrever-css>>. Acesso em: 20 jul. 2013.

Disponível em: <<http://tableless.com.br/pre-processadores-usar-ou-nao-usar/>>. Acesso em: 22 jul. 2013.

Disponível em: <http://pt.wikipedia.org/wiki/Cascading_Style_Sheets>. Acesso em: 25 jul. 2013.

Disponível em: <<http://www.w3c.br/Sobre>>. Acesso em: 25 jul. 2013.

Disponível em: <<http://lesscss.org/>>. Acesso em: 02 ago. 2013.

Disponível em: <<https://github.com/learnboost/stylus>>. Acesso em: 02 ago. 2013.