**Introdução**

Doador de sangue é o individuo que se voluntária para que seja feita e extração de seu sangue para que este seja utilizado no futuro em individuo especificado por esse ou quando surgir a necessidade.

Como sangue não se fabrica artificialmente se faz necessária a colaboração de doadores, pois a única fonte para esse material é um individuo doador, no Brasil a coleta, armazenamento e gerenciamento do sangue é feito por Hemocentros, Hospitais, Postos de Saúde, sejam estas de âmbito federal, estadual ou local.

Para se doar sangue é necessário fazer um cadastro previamente, exames para anemia e uma serie de aflições que se atestado positivos resultariam na desclassificação do voluntário como doador seja em caráter temporário ou permanente.

Atualmente o controle destes voluntários é feito por sistemas de escopo local da instituição onde o cadastro do mesmo foi feito seja por sistema informatizado ou registro em documentação física.

Consequentemente não existe nada que integre essas “ilhas” de doadores cadastrados, onde o banco de doadores entres diversos hemocentros e hospitais não compartilhado entre estes. Agilizar e facilitar esse processo consequentemente o tornando menos suscetível a erros é essencial, com uso de técnicas de desenvolvimento modernas isso se torna não somente possível mas também se torna algo mais fácil de ser resolvido de forma coerente e rápida

O desenvolvimento ágil de software é a metodologia de desenvolvimento a ser escolhida nesse caso, o desenvolvimento ágil segue uma serie de princípios fundamentais declarados em um manifesto.[[1]](#footnote-1)

Desenvolver software pode ser comumente comparado com a confecção de um trabalho de arte pois este requer criatividade em seu projeto e grande habilidade para atingir a sua compleição. O software é frequentemente maleavel porem incoerente e se mantém incompleto permanentemente. O desenvolvimento ágil de software é baseado em mudanças fundamentais no processo desenvolvimento de software visionadas a muito.

Desenvolvimento ágil significa uma mudança fundamental em como gerenciamos nossos projetos. Pois se um software totalmente funcional é o resultado em busca de qual trabalhamos, então então devemos monitorar nosso progresso baseado em que temos pronto e funcional no momento.

No desenvolvimento de software ágil nos mudamos nosso processo de gerenciamento do projeto para um no qual onde obtemos software funcional feito um pouco de cada vez. Desenvolvimento ágil derruba diversos paradigmas comparado com o modo de desenvolvimento que era utilizado até então, seguindo seus princípios fundamentais.

É necessário e essencial nos dias de hoje utilizar o desenvolvimento ágil de software, pois este traz mudanças na forma como o software é desenvolvido mudando paradigmas do processo de desenvolvimento que vem nos acompanhado por décadas, onde o desenvolvimento tradicional tem fases e checkpoints bem definidos durante seu processo de desenvolvimento onde no final de cada fase algum objetivo é cumprido e algo é entregue.

O desenvolvimento ágil tem iterações ao invés de fases onde no final de cada iteração temos código funcional que pode ser usado e avaliado, podendo responder aos requisitos do usuário que estão sempre mudando e evoluindo.

Em um projeto como o doando.se a capacidade de efetuar mudanças drásticas e a flexibilidade do software desenvolvido, podendo se adaptar a especificações em que ate mesmo os usuários finais deste serviço não estão certos de quais serão, é essencial.

Para o seu desenvolvimento com sucesso, e para produzir um resultado que tenha a capacidade de tirar parte do ineficiente trabalho burocrático existente nessas instituições de saúde, porem mantendo a capacidade para a inclusão facilitada de modificações futuras, torna o desenvolvimento ágil de software a metodologia a ser escolhida nesse tipo de projeto, ou em qualquer outro onde software funcional e satisfação e adequação aos requisitos do usuário são mais importantes do que a extensiva documentação de um software imutável e insatisfatório.

**2 Objetivo**

O objetivo é usar a tecnologia em prol de uma causa social, facilitando para as pessoas se voluntariarem na doação de sangue, criando uma ferramenta para os hemocentros reporem seus estoques de sangue através de uma tecnologia ágil, no intuito de facilitar todo o processo na comunicação entre doadores e hemocentros, para a resolução de um dos maiores problemas hoje enfrentada por hemocentros e hospitais. Em meio a esta questão a quem eles poderão recorrer? Televisão? Rádio? Tudo isso, hoje em dia, já está ultrapassado e há a ideia de levar todo o conteúdo e campanhas para a Internet e redes sociais.

Com esse intuito, adquire-se um foco para criar um web site chamado “*doando.se*” para sanar todos estes problemas, tais como o estoque baixo, a falta de sangue para certos pacientes, a urgência por bolsas de sangue, que muitas vezes levam a morte pela falta do mesmo.

A ideia e o objetivo principal é mostrar através de ferramentas ágeis que se pode obter sangue em um processo bem mais rápido que o normal, deixando claro que é necessário sempre da colaboração e conscientização da população em geral, tratando os casos com urgência com mais prioridade e obtendo agilidade através da ferramenta API do Google Maps, introduzindo uma busca feita pelo próprio hemocentro em um raio aproximado, busca-se o maior número de pessoas com o tipo sanguíneo especificado para entrar em contato e agendar a doação o mais rápido possível.

Após o cadastro ser efetuado, haverá total sigilo sobre os dados do doador e somente os hemocentros e hospitais terão acesso a eles, priorizando e garantindo a privacidade. Como forma de agilizar todo o processo de busca por doadores compatíveis, o número de pessoas que precisa ser utilizado em certos casos é essencial para não extrapolar a meta buscada.

Atualmente, no Brasil, a situação é drástica, pois, como exemplos, o estoque de sangue da Santa Casa de São Paulo está com a capacidade de 30%, em Belo Horizonte o número de doadores diminuiu 17%, em Curitiba 30%, no Rio de Janeiro caiu quase pela metade. Isso mostra que os doadores deixam de ir devido ao frio. No caso da professora Anna Maria Trapp, foi cancelada a cirurgia para a remoção de um tumor, toda a família se uniu e divulgou o problema na Internet, enfim, no outro dia ela conseguiu fazer a cirurgia. [1]

É seguindo o caso da professora que há o comprometimento em realizar este projeto ágil para sanar todos estes problemas de buscas por doadores, com este padrão de raciocínio, a interação com redes sociais trará mais expectativas para as pessoas necessitadas. De início, trataremos um processo de divulgação maior da web site com a ajuda dos próprios hemocentros, hospitais e pelas redes sociais, que é o maior fluxo de pessoas concentradas a todo o momento, focaremos também em um slogan, banners e frases criativas para atrair e induzir os usuários a se interessarem e se conscientizarem ao grave problema atual, pois vidas estão sendo perdidas e afinal, um dia pode ocorrer com um ente próximo, como um amigo ou parente.

Outro objetivo interessante é a visualização de pessoas cadastradas, cada usuário poderá se marcar e também ver quem é doador, através desse esquema, o boca a boca será essencial para aumentar o número de pessoas cadastradas.

A metodologia ágil para esse processo é transformar o curto tempo em qualidade do software, unindo a adaptação com a melhoria contínua. Uma das maiores prioridades é a interação com os possíveis doadores, mantendo a atenção total para todos, esclarecendo dúvidas frequentes, atualizando-se com novas campanhas, se empenhando completamente para a população. Outra metodologia ágil é a implementação de códigos com um número de linhas bem menor, simplificando todas as classes e métodos possíveis, eliminando todas as funções desnecessárias.

A coragem em implantar um suporte ideal para todo o processo de desenvolvimento com um bom relacionamento e uma boa comunicação com todos os tipos de pessoas, a aceitação e a avaliação em relação à mudança é essencial para obter uma continuidade no projeto e tudo isso mostra que é necessário unir a tecnologia ágil para atrair novas oportunidades para crescimento do site e priorizar o foco principal.

**3 Justificativa do projeto**

A escassez de bolsas de sangue pode custar uma vida em situações de emergência e tendo em vista que atualmente a única forma de se conseguir mais doadores em situação de necessidades é por contato pessoal e aleatório. É um fator agravante não apenas a busca de doadores em sí, mas sim, doadores com o tipo sanguíneo em questão.

A emergência ou doença que provoca a necessidade de transfusão de sangue, é de fato um problema individual, mas para a comunidade não é, pois de antemão todos sabemos que diariamente ocorrerão acidentes de trânsito, de trabalho, crimes violências e cirurgias.

Existem campanhas para doação de sangue em rádios, televisão, jornais, mas a grande verdade é que a atenção voltada para esses meios de comunicação, não é muito grande e por consequência deixam de saber sobre essas campanhas que estão acontecendo.

Não é novidade que o Brasil sofre por uma grande defasagem de doadores de sangue, todo mês cirurgias são canceladas pois hospitais não tem sangue suficiente em seus bancos de sangue para realizar as cirurgias. Um exemplo está numa matéria publicada pelo jornal Estadão(2012) onde diz que pelo menos 15 cirurgias foram canceladas no diâmetro das cidades que atinge Araçatuba, Marília e Jaú no interior de São Paulo.[1]

A internet nos dias de hoje, é o meio de comunicação mais rápido que existe, tudo que se é colocado na internet atinge milhões de pessoas instantaneamente. Pensando no poder de mídia que a internet possui, e na necessidade de divulgação sobre essas campanhas surgiu a ideia da construção de um aplicativo web que agrupe todos os possíveis doadores de sangue, mediante a um cadastro de uma forma simples, mas que contenham todas as informações que se necessita ter sobre o doador. Seria uma solução na hora em que uma pessoa necessitasse do auxílio das pessoas, a aplicação necessita ser de fácil uso para a população, e de fácil entendimento para os desenvolvedores na hora de uma possível alteração.

No desenvolvimento de software, os projetos são orientados a documentação que contém as especificações do software como suas funcionalidades, requisitos, restrições do software que são realizadas em uma fase onde o desenvolvedor conversa com o cliente para estabelecer essas regras, e a implementação do projeto, onde essas especificações são produzidas. Porém, muitas empresas não possuem recursos para processos de desenvolvimento pesados, para essa questão, foram criadas metodologias de desenvolvimento ágeis no final da década passada, para suprir as necessidades dos então atuais métodos de desenvolvimentos.

Metodologias de desenvolvimento ágil surgiu em 2001 quando especialistas em processos de desenvolvimento de software representando entre outros, os métodos Scrum e Extreme Programming (XP)   
é uma chamada para a necessidade de se desenvolver com rapidez, as questões de planejamento, implementação e custos e entrega do produto ao cliente, sempre foram um problema, gerando um conflito entre os desenvolvedores, na hora que o cliente efetuasse alguma mudança nos requisitos, toda documentação teria que ser mudada. Desenvolvimento ágil busca priorizar a comunicação entre o desenvolvedor e o cliente, preferindo conversas pessoais à outros meios de comunicação e tem as seguintes características:

* **Indivíduos e interações** ao invés de processos e ferramentas.
* **Software executável** ao invés da documentação.
* **Colaboração do cliente** ao invés de negociação de contrato.
* **Respostas rápidas à mudanças** ao invés de seguir planos. [2]

As chamadas metodologias de desenvolvimento tradicionais, devem ser aplicadas apenas para projetos em que não haverá muitas mudanças no decorrer da implementação, alterações onde a necessidade de refazer o código não for muito grande e não apresente um alto custo.

Ao se desenvolver um software é muito comum que os requisitos mudem com certa freqüência, enquanto ainda a implementação está sendo realizada, mudanças em toda documentação do projeto, levariam muito tempo pondo em risco a data de entrega ao cliente.

Nos dias atuais, tudo tem que ser produzido o mais rápido possível, em questão ao desenvolvimento ágil, existem hoje no mercado, ferramentas de desenvolvimento que trabalham com condensação de códigos to, por exemplo: um método que antes era escrito com trinta linhas de código, passa a ser escrito com quinze linhas, sendo uma alternativa rápida para o desenvolvimento, fazendo com o que o programador use um tempo menor digitando as linhas de comando e também na hora de fazer uma possível alteração no código fonte, o programador teria que lidar com menos linhas de códigos a serem analisadas, sendo assim, a manutenção se torna muito mais prática e rápida.

Por ser uma maneira atual e mais prática de se desenvolver software, a implementação ágil evita vários custos na implementação de um projeto e apresenta grandes vantagens na hora de se escrever um programa, vantagens como agilidade, o tempo gasto na construção da documentação é usados em fins mais produtivos foi usado a metodologia de desenvolvimento ágil na construção dessa aplicação.

**4 Fundamentação Teórica**

A Web 1.0, era uma plataforma, onde usuários, no caso espectadores, podiam apenas navegar entre as páginas e visualizar seu conteúdo; não sendo permitido nenhum tipo de interação com o serviço, como por exemplo, reeditar ou alterar algum conteúdo exposto na página.

        Além disso, a grande maioria de serviços prestados na Internet, eram liberados mediante a pagamentos, ou seja, para usufruir de algum conteúdo, era necessário realizar um pagamento e uma licença única e restrita àquele usuário era gerada para que o mesmo tivesse acesso a tais serviços.

         Apesar do vasto conteúdo de informações transmitidas para o usuário, a Internet era aberta para todos, sem  nenhum dono para controlar os dados acessados.

Eis que surge a segunda fase da Internet que foi o marco da evolução da Internet, onde Tim O’Reilly no ano de 2005 diz as seguintes considerações:

*A web 2.0 é a mudança para uma Internet como plataforma, e um entendimento     das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva. (REILLY, 2005). [5]*

      Nesta mudança, obteve-se um enorme avanço em relação à interação do usuário, publicações de conteúdo, edições, criações de páginas online sem nenhuma dificuldade em nível de programação e também pode-se citar que o utilizador disponibilizava servidores gratuitamente.

Blogs, PodCast, Hi5, De.li.cio.us são alguns termos que podem ser citados como exemplo da grande variedades de serviços que estão hoje disponíveis graças ao avanço que a Web 2.0 propôs. [6]

       Outras características importantes, são as informações que se atualizam constantemente a fim de diminuir e obter correções para maior comodidade do utilizador, interações com pessoas do mundo inteiro, nisso, incluímos aplicações como Messenger, Googletalk, Skype e Voip, algumas ferramentas de escrita online como Google Docs e Podcast e o avanço do acesso a vídeos que cresceu muito nos últimos anos com o surgimento do Youtube, GoogleVideos e YahooVideos.

Sendo assim, a demanda de programação para a web foi crescendo cada vez mais, e problemas como prazos de entrega que não eram respeitados por vários fatores determinantes e até mesmo o não entendimento por parte da parte de programação que ficava um pouco confusa na hora de ser feita a interpretação dos textos da documentação, surgiu um novo modo de ser feita a organização dessas documentações, algo que poria a prática na implementação do software em algo muito mais eficiente e com bastante clareza que foi o manifesto agile. (colocar definição no glossário)

Antes do manifesto, vários fatores agravavam o não cumprimento de metas e prazos de entrega dos softwares aos clientes, tais como poucas reuniões com os clientes, onde ficavam faltando informações sobre o projeto, advogados que brigavam por causa de contratos, documentação muito extensa que acarretava o desentendimento dos desenvolvedores na hora de analisar a documentação

O manifesto agile, que significa desenvolver com agilidade teve como precursor os próprios desenvolvedores ao tentar dominar um pouco o processo produtivo nas empresas em que trabalhavam, não aceitando as regras impostas de algum gerente que muitas vezes não sabiam como funcionava o processo da fábrica de software, então eles começaram a mudar a maneira como trabalhavam, fazendo o que achavam que seriam melhor para os projetos em que trabalhavam e a partir dessas ideias, outros desenvolvedores se juntaram ao manifesto.

O manifesto agile tem como seus valores, quatro itens. Sendo eles:

* Os indivíduos e suas interações acima de processos e ferramentas.
* O software em funcionamento acima da documentação abrangente.
* A colaboração dos clientes acima da negociação de contratos.
* A capacidade de resposta a mudanças acima de se seguir um novo plano.

A expressão “Metodologias Ágeis” tornou-se conhecida em 2001, quando especialistas em processos de desenvolvimento de software representando entre outros, os métodos Scrum e Extreme Programming (XP), foram estabelecidos princípios e características comuns destes métodos. Assim foi criada a “Aliança Ágil” e efetuou-se o estabelecimento do “Manifesto Ágil”.

Extreme Programming (XP) é uma metodologia ágil para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e que se modificam rapidamente, sendo diferente dos métodos de desenvolvimento comum o constante feedback, a obordagem incremental e o encorajamento da comunicação entre pessoas.

Muitas de suas regras causam estrenheza em seu primeiro contato ou até mesmo não fazem sentido se aplicadas isoladamente, porém é a porça de seu conjunto que sustenta o sucesso da XP, trazendo a verdadeira revolução no desenvolvimento de software, sendo como seu objetivo dar agilidade ao processo de desenvolvimento e buscando sempre garantir a satisfação do cliente. Suas práticas são conduzidas por quatro princípios básicos, sendo eles:

1. **Princípio de comunicação** – busca manter o relacionamento com o cliente sempre da melhor forma possível, com reuniões práticas e esclarecedoras, prezando pelo contato pessoal ao invés de outros meios de comunicação.
2. **Princípio da simplicidade** - busca-se alcançar o objetivo do projeto de software com o menor número possível de classes e métodos, também está em sua caracterísitca a procura da implementação de requisitos atuais, evitando adicionar funcionalidades que podem ser importante apenas no futuro. A aposta do XP é fazer algo simples hoje do que implementar algo complicado hoje que talvez não venha a ser usado.
3. **Princípio do feedback** – o desenvolvedor terá constantes feedback do código que está sendo construiído se informando num geral sobre todo o projeto e recebendo também feedback do cliente para qual o projeto é destinado. A informação do código é conseguida através de constantes testes em que indicam os erros que o código possui, tanto erros pessoais quanto erros de código integrado.
4. **Princípio da coragem** – Sabe-se que nem todas as pessoas tem facilidade de comunicação e possuim bom relacionamento interpessoal, sendo um método voltado para a simplicidade, a equipe pode experimentar a buscar novas soluções, além disso, é preciso coragem para obter e cobrar constantes feedback do cliente. [referência]

As práticas do modo XP começam pelo planejamento, sendo definido o que é, e o que não é necessário ser feito no projeto, baseando-se em requisitos atuais, deixando que requisitos futuros sejam implementados somente quando haja a necessidade deles no projeto. A entrega do software acontece conforme os requisitos vão sendo implementados, a cada requisito há uma atualização do software, cada versão entregue ao cliente deve conter o menor tamanho possível, contendo os requisitos de maior valor para o negócio.

A descrição do projeto fica por conta de metáforas, sendo assim, a descrição não conta com termos técnicos, o que possibilita um maior entendimento para o cliente, sendo muito importante a participação do cliente em todo o processo de desenvolvimento para sanar dúvidas de requisitos, evitando atrasos ou até mesmo construções erradas.

A Extreme Programming (XP) prioriza a validação do projeto durante todo o processo de desenvolvimento. Os desenvolvedores implementam o software criando primeiramente os testes, sendo a implementação de código feito em duplas, ou seja, dois desenvolvedores trabalham em um único computador, procurando identificar erros sintáticos e semânticos, pensando em conjunto em como melhorar o código que está sendo implementado.

Muito importante também é a refatoração do código, tornando um código lapidado, limpo, para que não haja problema na hora que for feita a leitura deste código que perterce a todos os membros da equipe. Isso significa que qualquer pessoa pode perceber alguma necessidade no código, pode fazê-lo mesmo sem ter sido a pessoa propriamente dita quem desenvolveu aquele código, mas desde que faça os testes necessários e não prejudique as funcionalidades atuais. A grande vantagem de todos ter acesso a todo o código, é que se algum membro abandona o projeto, outro membro não terá grandes dificuldades ao se encarregar das tarefas que foram deixadas pelo outro membro do projeto, pois todos conhecem todas as partes do software mesmo que não seja de forma detalhada.

A XP também assume, que não se deve trabalhar mais que 40 horas semanais. Caso seja necessário trabalhar mais que 40 horas, significa que existe um problema sério no projeto que deve ser resolvido não com o aumento de horas trabalhadas, mas sim como melhorar o planejamento. [referência]

Além da evolução em como a internet era vista pelos navegadores, após a evolução da web 1.0 para web 2.0 como citado acima, o desenvolvimento também sofreu uma evolução, a maioria dos sites antigos eram todos escritos em HTML.

HTML foi projetado para definir a estrutura de um documento da web, onde o código <p> é um elemento estrutural que se refere à "parágrafo", <LI> também é um elemento estrutural que se refere à "listar item" enquanto o HTML expandia, mais elementos foram adicionados. Estes elementos, definem como o texto que será exibido, será formatado.

Em outras palavras, forma e conteúdo se tornaram inseparáveis em HTML, porém o XML foi projetado para fazer exatamente isso onde <title> não define a forma e sim o conteúdo, o mesmo com <link> e <description>, assim os dados podem ser exportados livres de restrições de formatação.

Com a forma separada, usuários não precisavam saber linhas de código para publicar informações na internet, nascem então os blogs, onde a programação já era toda definida pelo servidor em que hospedava os blogs, o usuário a partir de sua conta, somente tinha que enviar informações para ser exibida.

Mesmo sendo um avanço tecnológico grande, para se construir um site, o programador dedicava muito tempo do seu tempo para digitar inúmeras linhas de comando, as vezes para se construir pouca coisa, necessitava de um grande número de linhas.

Com o aprimoramento das tecnologias, novas linguagens de programação para web foram surgindo, e que satisfazem essa questão da necessidade do programador digitar várias linhas de comando. Exemplo dessas tecnologia de desenvolvimento ágil é a linguagem Ruby que foi escolhida por ser uma linguagem dinâmica e atual, com constante crescimento no mercado de trabalho, segue abaixo suas definições e ferramentas que podem ser usadas juntamente com a linguagem.

Ruby é uma linguagem de programação interpretada multiparadigma, de tipagem dinâmica e forte, com gerenciamento de memória automático, originalmente planejada e desenvolvida no Japão em 1995 por Yukihiro “Matz” Matsumoto, para ser usada como linguagem de script. Matz queria uma linguagem de script que fosse mais poderosa do que Perl e mais orientada a objetos do que Python.

A linguagem também suporta programação funcional, orientada a objetos, imperativa e reflexiva. Foi inspirada principalmente por Python, Perl, Smalltalk, Eiffel, Ada e Lisp, sendo muito similar em vários aspectos a Python. Atualmente, Ruby é a 11ª (décima primeira) linguagem de programação mais popular do mundo, de acordo com o Índice Tiobe [usar referencia do tcc]

Ruby on Rails é seu framework, foi escrito por David Heinemeier Hansson em Ruby, organizado em MVC (Model-View-Controller). Lançado para o público em 2004, sua característica é a facilidade na hora de programar, tornando códigos menores o que leva o programador programar mais em menos tempo.

Este framework surgiu para trazer uma forma mais dinâmica para o desenvolvimento de sites orientado a banco de dados, focando na agilidade para aumentar a produtividade em uma quantidade de linhas menores.

Esta é uma plataforma com opinião, ela impõe que tem um modo certo de se fazer as coisas e foi desenvolvido para encorajar o programador a seguir este modo.

Rails segue algumas filosofias, e uma delas é denominada DRY (Don't Repeat Yourself) ela diz que não é bom que não é bom escrever o mesmo código mais de uma vez.

Em uma plataforma de arquitetura MVC (Model-View-Controller) o ROR utiliza um recurso chamado Scaffolding que gera automaticamente controllers e views a partir de um certo modelo mostrado para o programador. RoR também provê abstração de SGBDs (Sistema de Gerenciamento de Banco de Dados), o usuário usa apenas uma sintaxe para MySQL, PostgreSQL, SQL Server, Oracle, Firebird, entre outros

O Rails faz suposições sobre o que você quer fazer e como você estará fazendo isto, em vez de deixá-lo mudar cada minúscula coisa através de intermináveis arquivos de configuração. [9]

Esta ferramenta poderosa de programação para Web pode ser rodado em Windows, Linux e Mac OS e a permissão para o uso do software é gratuitamente, podendo modificar sem restrição, com direitos a usar, copiar, distribuir e até vender.

A linguagem disponibiliza o uso do coffescript que é uma pequena linguagem de script que compila para JavaScript para desenvolvimento de websites, é apenas JavaScript, compilando de um-para-um para o js (JavaScript) equivalente.. CoffeScript tenta mostrar as boas partes do JavaScript em um modo simplificado. Esta linguagem converte os scripts .coffe em scripts .js (JavaScript). [10] Segue abaixo exemplos de sintaxes escrita em JavaScript e a mesma sintaxe escrita em CoffeScript.



Além do CoffeScript, o uso do HAML também é utilizado. A idéia de HAML é somente a estética da sintaxe, não funcionalidades. É uma sintaxe simplificada do HTML, renderizado para HTML que não necessita o fechamento de tags. Segue abaixo exemplo de sintaxe escrita em HTML e a mesma escrita em HAML.



Quando o assunto diz respeito a CSS, a tecnologia utilizadada chama-se Sass.que é uma extensão para o CSS3, desenvolvida [Nathan Weizenbaum](http://en.wikipedia.org/w/index.php?title=Nathan_Weizenbaum&action=edit&redlink=1), adicionando regras aninhadas, variáveis, mixins, herança de seletor. É traduzida para o formato CSS padrão. [11]



**5 Estudo de caso**

1. "Manifesto for Agile Software Development." 2003. 24 Jun. 2012 <<http://agilemanifesto.org/>> [↑](#footnote-ref-1)