

# Estudo Comparativo entre Ferramentas de Geração de Aplicativos para Dispositivos Móveis

Henrique Carvalho Parreira \*

Pasteur †

16 de novembro de 2015

## Resumo

Com a onipresença da internet na vida das pessoas e do mercado, vem aumentando e se tornando exigência web sites para empresas de todos os ramos e tamanhos. Em razão disso, diariamente empresas gastam valiosos recursos no desenvolvimento de web sites para seus produtos e negócios. Com a popularização dos *smartphones*, vem surgindo uma nova necessidade, a de desenvolvimento de aplicações *mobile* para suprir essa demanda de uma maior presença nas diferentes mídias. O barateamento das ferramentas para criação automática de web sites e aplicações *mobile* facilita a inserção na internet dessas empresas, entretanto o resultado acaba sendo de mal qualidade. A motivação deste trabalho é apresentar 3 (três) ferramentas distintas de criação e geração de aplicações móveis: AppGenerator, WebFlow e AppMachine. O objetivo é demonstrar como as ferramentas se diferenciam comparando 3 (três) métricas: a velocidade no desenvolvimento, qualidade e tamanho do código gerado e o número de linhas resultantes. Juntamente com as métricas que serão comparadas ao final dos testes será feito um questionário com os usuários afim de validar a experiência utilizando cada uma das ferramentas para comparar com os resultados anteriores. Os resultados mostram que a verticalização na criação do web site e aplicação mobile que a ferramenta AppGenerator utiliza, tem no final um resultado mais conciso, de melhor qualidade e de maior velocidade no desenvolvimento da aplicação final, com uma média de 3 (três) vezes em relação as outras ferramentas utilizadas para comparar, e um código com um número de linhas menor em 3,6 vezes quando comparado.

**Palavras-chaves:** App, aplicação web, gerador.

## 1 INTRODUÇÃO

Com a onipresença da internet na vida das pessoas e do mercado, vem aumentando e se tornando exigência web sites para empresas de todos os ramos e tamanhos (IBGE, 2015). Existe um grande interesse em ferramentas que auxiliam a criação e desenvolvimento de aplicações web (GLOBO, 2015). A maioria dessas ferramentas auxiliam os web *designers* a desenvolverem de forma automatizada paginas estáticas, ou seja, paginas web cujo conteúdo é principalmente texto e imagens, sem módulos informativos mais interativos.

---

\*henrique@citspot.com

†pasteur@pucminas.br

Basicamente essas ferramentas permitem a esses profissionais selecionarem estilos de *CSS*<sup>1</sup>, uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, e a ferramenta gera o site baseado no estilo definido.

A fácil implementação e geração de web sites é de extrema importância para o mercado atual (VERGE, 2014), mas a simples geração de páginas estáticas não é mais suficiente (EMPRESAS, 2015). Hoje é necessário a geração de web sites mais complexos, contendo características e informações importantes, como página criada de maneira mais simples e rápida, a aplicação precisa ser otimizada para todas as telas disponíveis dentre elas: monitores de várias resoluções, *tablets* e *smartphones*, a página criada não pode ser estática e precisa ter módulos mais complexos e informativos, e com o advento dos *smartphones* e *tablets*, surgiu a necessidade das aplicações web funcionarem perfeitamente nesses novos dispositivos.(CRUNCH, 2015)

## 1.1 Objetivos do Trabalho

Esse artigo tem como objetivo principal apresentar 3 (três) ferramentas de desenvolvimento de aplicações de web sites e aplicativos móveis: *AppGenerator*, *WebFlow* e *AppMachine*. Essas ferramentas tem como resultado final aplicações móveis, mas se diferem na maneira com que essa aplicação é gerada. Todas as ferramentas tem como objetivo facilitar o desenvolvimento de aplicações web e aplicações *mobile* e foram criadas para serem sistemas *web*, funcionando em todos os navegadores modernos, afim de maximizar a sua utilização por *designers* e empreendedores.

Para melhor avaliar as ferramentas foi escolhida 3 (três) medidas de avaliação para demonstrar de várias maneiras as principais características no desenvolvimento de aplicativos móveis, esses métodos são os mais utilizados para realizar comparações quando se fala em ferramentas voltadas para usuário final. O primeiro método é a contagem do número de linhas, dessa forma poderá ser quantificado qual o resultado final tem maior organização e mais facilidade na manutenção. A segunda forma de comparar as ferramentas será fazer um comparativo de tempo de desenvolvimento da aplicação final entre usuários utilizando as ferramentas, com o objetivo de verificar qual ferramenta é mais rápida de se utilizar. Por último será analisado o tamanho em *bytes* da aplicação resultante.

## 1.2 Justificativa

Com tantas maneiras de desenvolver aplicações *web* e *mobile*, é importante que os usuários que fazem uso de ferramentas para esse fim, saibam como é o funcionamento e mais importante qual o resultado encontrado em cada uma delas. A utilização dessas ferramentas por parte de usuários leigos, nesse artigo, aqueles usuários sem conhecimento em linguagens de programação é muito grande, e por isso esse artigo foi feito com o intuito de exemplificar 3 (três) tipos distintos de ferramentas, e mostrar resultados comparativos das impressões de usuários reais dessas ferramentas.

A motivação no desenvolvimento desse artigo é que após sua leitura esperasse seja possível entender alguns dos tipos de ferramentas de geração de aplicações, e que o leitor tenha senso crítico e conhecimento para escolher a melhor para o objetivo que esteja buscando.

---

<sup>1</sup> CSS - Cascading Style Sheets

### 1.3 Organização do Trabalho

Na seção 2, será feita uma contextualização dos tipos de ferramentas de geração de aplicações e será listado as 3 ferramentas utilizadas nesse trabalho e suas características, como funcionamento, arquitetura do sistema com sua respectiva explicação e interface de uso. A seção 3, metodologia, será explicado as etapas que foram utilizadas para desenvolver o trabalho, desde a criação da ferramenta, a seleção de ferramentas que possam ser comparadas, a definição dos testes e como esses serão realizados. Na seção 4 do artigo será descrito os requisitos da aplicação base que será utilizada para os testes a serem feitos com as ferramentas. A seção 5, avaliação, onde será reunido os resultados encontrados e por fim a conclusão do trabalho contendo as impressões encontradas e trabalhos futuros. E por fim, na seção 6, conclusões, esta contida as dificuldades encontradas no trabalho e trabalhos futuros a serem realizados.

## 2 FERRAMENTAS PARA GERAÇÃO DE APLICAÇÕES

Existem vários tipos de ferramentas utilizadas para o desenvolvimento de aplicações sem a necessidade de conhecimento prévio em linguagens de programação. Existem ferramentas que fazem uso intenso de *drag and drop*, onde o usuário "arrasta" elementos pré definidos que gostaria de utilizar para um *canvas* de produção. Outro tipo de ferramenta são as que funcionam selecionando opções pré definidas dadas pelo sistema onde ao final do processo a aplicação é gerada. E finalmente existem ferramentas que utilizam de blocos funcionais que representam pseudocódigo que tem como objetivo simular o desenvolvimento *hardcore* porém de forma mais acessível a usuários sem grandes conhecimentos em linguagens de programação.

Para esse trabalho será escolhido uma ferramenta de cada tipo citado anteriormente afim de comparar melhor os vários tipos de ferramentas disponíveis. A ferramenta *AppGenerator* é uma ferramenta que fornece aos usuários escolhas pré determinadas, onde ao final do processo será gerada a aplicação final. O sistema *web Webflow*, uma plataforma semântica de vanguarda para publicação pessoal, com foco na estética, nos padrões Web e na usabilidade(??), será utilizada para representar ferramentas que fazem uso de elementos de interface e funciona da forma *drag and drop*. O aplicativo *AppMachine* é um software para fazer aplicativos nativos para *iPhone* <sup>2</sup> e *Android* <sup>3</sup> ([APPMACHINE, 2013](#)), será utilizado para a representar ferramentas que utilizam de blocos de pseudocódigo para criar a aplicação final.

Importante ressaltar que nenhuma das ferramentas utilizadas nesse artigo geram código fonte editável pelo usuário, mas sim geram uma aplicação funcional ao final do processo. A ferramenta *AppGenerator* gera a aplicação final através de uma aplicação denominada neutra, que vai preencher o aplicativo com o estilo selecionado durante a utilização da ferramenta. A ferramenta *WebFlow* gera a aplicação final, através de pedaços de código pré definidos e sem acesso ao usuário final. Cada novo elementos que for selecionado pelo usuário tem um conjunto de código relacionado que é escrito em um arquivo e ao final do processo, gera a aplicação final. A ferramenta *AppGenerator* funciona através de blocos funcionais de pseudocódigo que ao serem utilizadas escreve em um arquivo o código relacionado ao bloco e ao final do processo gera a aplicação final.

<sup>2</sup> Aparelho smartphone da empresa norte americana Apple

<sup>3</sup> Sistema operacional mobile da empresa norte americana Google

## 2.1 AppGenerator

O *AppGenerator* é uma ferramenta web, funcionando em todos os navegadores modernos, desenvolvida afim de oferecer para o web *designer* uma forma mais simples e eficaz de criar aplicações web e *mobile* nas plataformas iOS <sup>4</sup>, da empresa Apple <sup>5</sup>, e *Android*, da empresa Google. O *AppGenerator* funciona em 3 (três) etapas distintas. A primeira é a captura de requisitos do usuário, onde é capturado as especificações e estilo do aplicativo fornecendo ao usuário uma gama de escolhas para o desenvolvimento da aplicação final. A segunda etapa é onde é salvo essas informações no banco de dados para futuras requisições. E a terceira e ultima etapa é onde o usuário requisita os aplicativos das plataformas que gostaria de gerar é o sistema busca no banco de dados as especificações cadastradas.

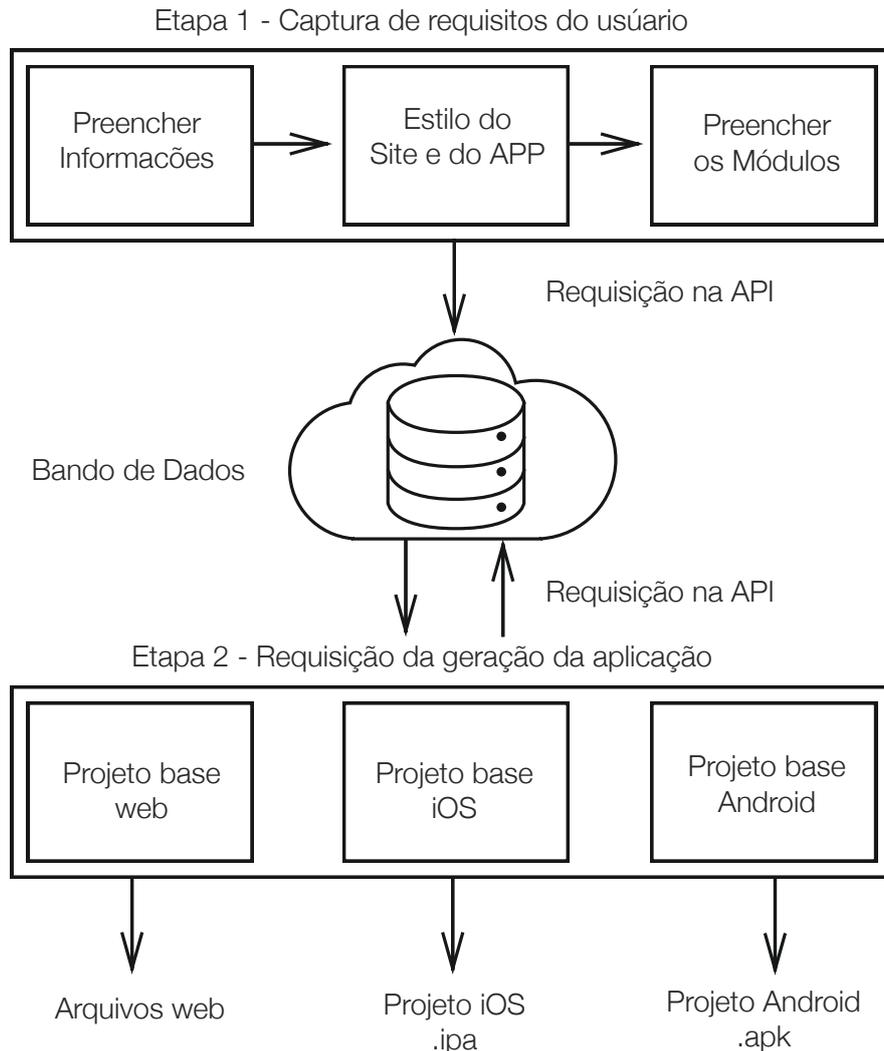
A arquitetura do *AppGenerator* é simples, tendo como base o banco de dados modularizado, permitindo escalar para novas funcionalidade e escolhas sem dificuldades. Segue na figura 2.1 a arquitetura do *AppGenerator* e a explicação de cada parte do modelo.

---

<sup>4</sup> Sistema operacional mobile da empresa norte americana Apple

<sup>5</sup> Empresa de tecnologia norte-americana

Figura 1 – Arquitetura do AppGenerator.



Etapa 1 - Captura de requisitos do usuário.

A interface do *AppGenerator* é simples e objetiva, com o objetivo de facilitar e tornar rápido o desenvolvimento de aplicações web e *mobile*. A etapa 1, na figura 2.1, de captura de requisitos do usuário é toda feita nessa interface web, é tem como objetivo reunir todas as informações e especificações que a aplicação irá receber. A etapa 1 é dividida em 3 subdivisões, onde ao final do processo de coleta, essas informações serão salvas no banco de dados do sistema. As 3 subdivisões são:

- Etapa 1.1 - Preencher informações: Nessa etapa o usuário preenche o conteúdo inicial da sua página. Esse conteúdo contempla: descrição do negócio, texto de apresentação, o link para as redes social *facebook*<sup>6</sup> e *twitter*<sup>7</sup>, telefone, web site já existente e informações relativas ao desenvolvimento de sua aplicação.

<sup>6</sup> Rede social norte americana

<sup>7</sup> Rede social e um servidor para microblogging norte americano

- Estilo do site e do app: Na escolha do estilo é fornecida uma gama de opções que a aplicação final irá receber. As escolhas são limitadas as cores do conteúdo, estilo e cores do menu de navegação, cores dos detalhes da aplicação. O número de opções ofertada é limitada para facilitar e tornar o desenvolvimento mais rápido. Quando a aplicação for gerada, após a etapa 2, alguns detalhes, como paleta de cores respectiva do sistema operacional *mobile*, que não foi fornecida a opção de mudança serão definidas baseada em padrões web observadas, e guias de uso fornecida pelas empresas Apple e Google.
- Preencher os módulos: Os módulos são funcionalidades definidas e desenvolvidas para aumentar as informações presentes na aplicação gerada. Atualmente a ferramenta possui 7 módulos, onde o web *designer* seleciona qual deles gostaria de fazer uso. Esses módulos são: cardápio, tabela de preço, eventos, quartos, produtos, comentários, galeria de fotos.

A figura 2.1 é uma tela da interface web do *AppGenerator* exemplificando o desenvolvimento de uma aplicação. Essa tela é relativa a etapa 1.1 da figura 2.1, mostrando a captura de requisitos do usuário de forma mais específica, a captura das informações contextuais do aplicativo que será gerado ao final.

Figura 2 – Tela da etapa 1 - Captura de requisitos do usuário.

The screenshot shows the 'citspot' web interface. At the top, there is a search bar with the text 'Procure aqui a categoria ou o nome de cidade' and a 'Cadastrar Spot' button. Below the search bar, the interface is split into two main sections: '2 Informações' and '3 Personalização'.

**2 Informações**

- Telefone para Contato
- E-mail para contato
- Endereço de Site
- Facebook
- Twitter
- Tipos de Restaurante: Buffet / Por pessoa, Rodizio, Self Service / Por kilo, À La Carte
- Recomendado para: Amigos, Casais, Crianças, Família, Homossexual, Idoso, Infraestrutura para Deficientes
- Tipos de comida: Alemã, Argentina, Armena, Australiana, Brasileira, Chinesa, Coreana, Cubana, Egípcia, Escocesa, Espanhola, Francesa, Frutos do Mar, Grega, Indiana, Inglesa, Internacional, Iraniana, Irlandesa

**3 Personalização**

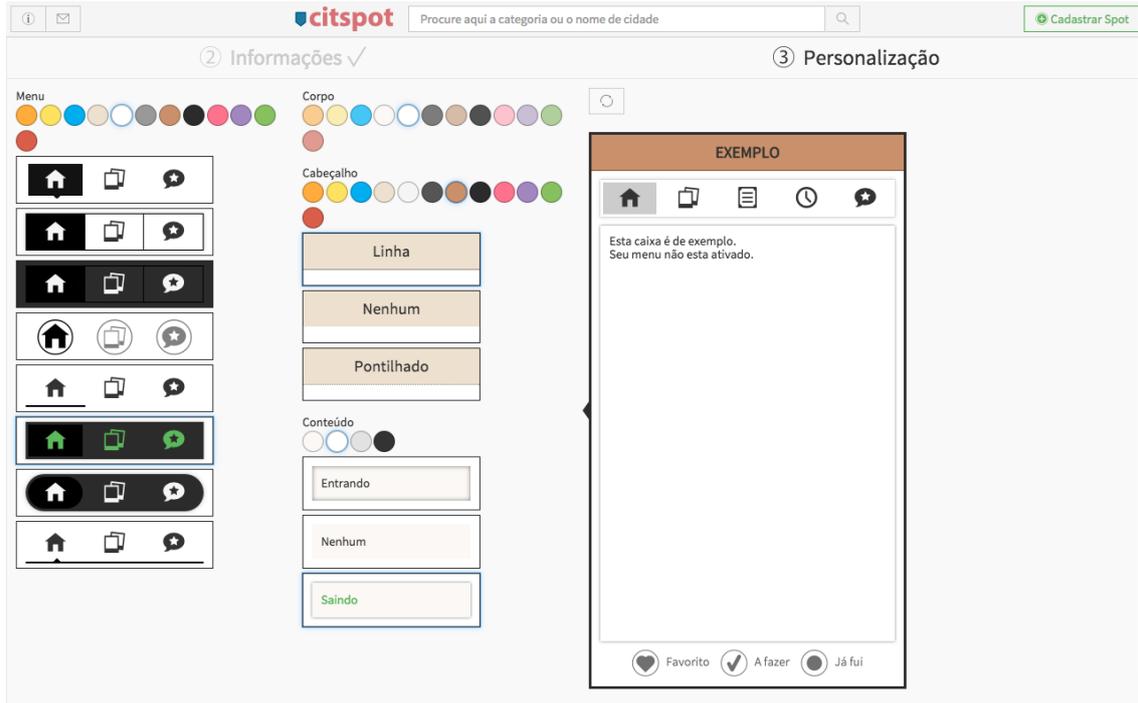
- Formas de Pagamento: American Express, Dinners, Dinheiro, Discover, JCB, MasterCard, Visa
- Cheque
- Horário de funcionamento table:

Dia	Início	Fim
Domingo	00:00	00:00
Segunda-feira	00:00	00:00
Terça-feira	00:00	00:00
Quarta-feira	00:00	00:00
Quinta-feira	00:00	00:00
Sexta-feira	00:00	00:00
Sábado	00:00	00:00

- Ambiente: Aconchegante

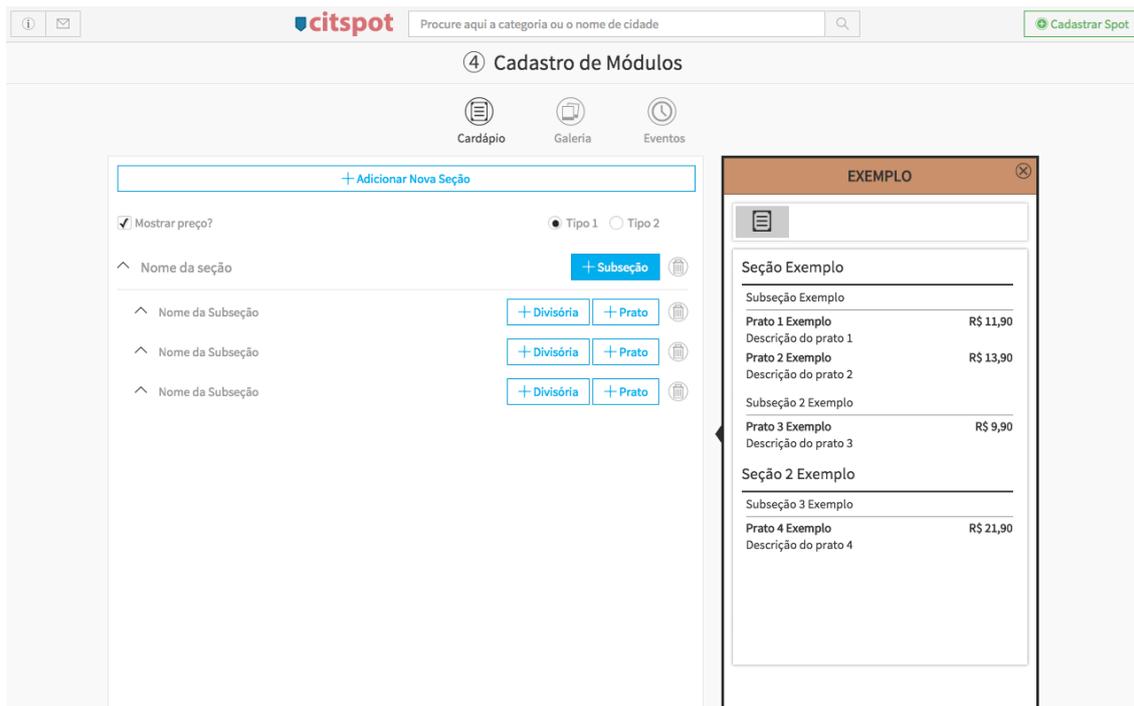
A figura 2.1 é uma tela da interface web do *AppGenerator* que é relativa a etapa 1.2 da figura 2.1, mostrando o preenchimento do estilo do aplicativo que será gerado ao final do processo. Como exemplificado na figura, nessa etapa o usuário seleciona o estilo de menu e cores do aplicativo final. Essas escolhas serão salvas no banco de dados e ao final do processo, serão requisitadas para montar o aplicativo final.

Figura 3 – Tela da etapa 2 - Captura de requisitos do usuário.



A figura 2.1 é uma tela da interface web do *AppGenerator* mostrando a terceira e última etapa do processo, na figura 2.1 é a etapa 1.3. Nessa etapa é onde o usuário preenche todas as informações relativas ao aplicativo que está sendo gerado, e os módulos que irão ser preenchidos são baseados no tipo de aplicação que o usuário deseja, por exemplo: No caso da criação de um aplicativo móvel para um restaurante, os módulos que serão exibidos para preenchimento são os de cardápio, galeria de fotos, eventos e comentários. Todas as informações preenchidas serão salvas no banco de dados, e ao final do processo, essas informações serão requisitadas para preencher o aplicativo final.

Figura 4 – Tela da etapa 3 - Captura de requisitos do usuário.



Descrição e funcionamento do Banco de dados:

O *AppGenerator* se apoia fortemente no banco de dados modularizado, onde é possível expandir qualquer uma das escolhas e funcionalidades de forma independente do resto do sistema. Essa característica permite a aplicação final ser menor e mais eficiente. No banco de dados ficarão todas as informações coletadas na etapa 1 - captura de requisitos do usuário, e serão utilizadas para gerar a aplicação final.

Na figura 2.1 a API <sup>8</sup> para acesso ao banco de dados, é utilizado uma API que foi desenvolvida afim de facilitar principalmente a geração da aplicação mobile. A través da API é requisitada todos os estilos e informações preenchidas pelo web designer através de uma URL <sup>9</sup>. O retorno dessa URL é um JSON <sup>10</sup>, uma formatação leve de troca de dados, contendo todas as informações específicas durante o processo de criação. Fazendo uso da API é possível isolar o banco de dados do restante da aplicação podendo ser utilizados e acessados por todas as aplicações tanto web quanto *mobile*, e tornar as requisições a esse banco mais eficientes.

Etapa 2 - Requisição da geração da aplicação.

A etapa 2 especificada na figura 2.1 tem como objetivo proporcionar ao usuário escolher quais plataformas gostaria de ter seu aplicativo gerado, podendo escolher entre aplicativo web, aplicativo *mobile* iOS e aplicativo *mobile* Android. Isso é possível devido aos arquivos base de cada aplicação, que na figura 1 são chamados de projetos base. Os projetos base são um conjunto de arquivos contendo a estrutura básica de cada aplicação. Após escolhido pelo usuário as plataformas que gostaria de gerar sua aplicação, os arquivos base são gerados e fazem uma requisição ao banco de dados, utilizando a API, afim de

<sup>8</sup> API - Interface de Programação de Aplicações

<sup>9</sup> URL - *Uniform Resource Locator*, traduzindo: Localizador Padrão de Recursos

<sup>10</sup> JSON - *JavaScript Object Notation*, traduzindo: Notação de Objetos JavaScript

requisitar os estilos definidos pelo usuário na etapa 1. As informações retornadas da API irão preencher esses arquivos bases, e ao final do processo o resultado é a aplicação nas plataformas escolhidas.

- Arquivos web - Como resultado final da aplicação web, todos os arquivos necessários são gerados e organizados da melhor forma possível garantindo uma boa eficiência e alta reutilização e manutenção de código. Os arquivos de cada página, assim como os arquivos de CSS e de javascript <sup>11</sup>. A estrutura da página e dos arquivos tem uma base pré-definida que ajuda na criação de um bom código, portanto o conteúdo dessa página, tem alguns campos preenchidos que vieram através da requisição da API. Esses campos são os de estilo que ficam no arquivo *.css*.
- Projeto iOS *.ipa* <sup>12</sup> - Para a aplicação iOS também são gerados cada um dos arquivos necessários para o aplicativo, organizados nas respectivas *view* e *controllers*. Além do logo do app. O conteúdo do aplicativo, assim como seu estilo é conhecido através da chamada da API que trouxe do banco todas essas informações.
- Projeto *Android .apk* <sup>13</sup> - A aplicação Android tem os arquivos muito similares dos gerados pela aplicação iOS, contendo as *views* e *controllers*, e seguindo as recomendações de criação de aplicativos para o Android, como os estilos de cores e formato dos botões.

## 2.2 WebFlow

O *WebFlow*, uma plataforma semântica de vanguarda para publicação pessoal, com foco na estética, nos Padrões Web e na usabilidade(??), será utilizado para criar a aplicação web. A ferramenta gera através de *templates* <sup>14</sup> pré-determinados é possível criar um web site. O resultado final gerado pelo *WebFlow* é um site estático e funcional. A principal diferença do *WebFlow* para o *AppGenerator*, é a forma como é feita o desenvolvimento da aplicação web, através de *template* pré-determinado, ao invés de permitir um número de escolhas de estilo, como é feita no *AppGenerator*. Outras diferenças importantes que podem ser citadas, é o fato de que no *WebFlow* não é gerado uma aplicação *mobile* nativa, mas sim, uma adaptação responsiva do site, termo utilizado para definir um site que tem seu estilo modificado dependendo do tamanho de tela, dessa forma a experiência do usuário e resultado final são afetados negativamente.

O funcionamento da ferramenta *WebFlow* é bem simples e direta, com foco no canvas da aplicação. O desenvolvimento de uma aplicação é bem direto, onde o usuário escolhe um *template* e a partir desse momento ele dá início ao desenvolvimento do aplicativo móvel. A seguir segue uma imagem contendo a arquitetura do sistema e uma explicação de cada uma das etapas.

---

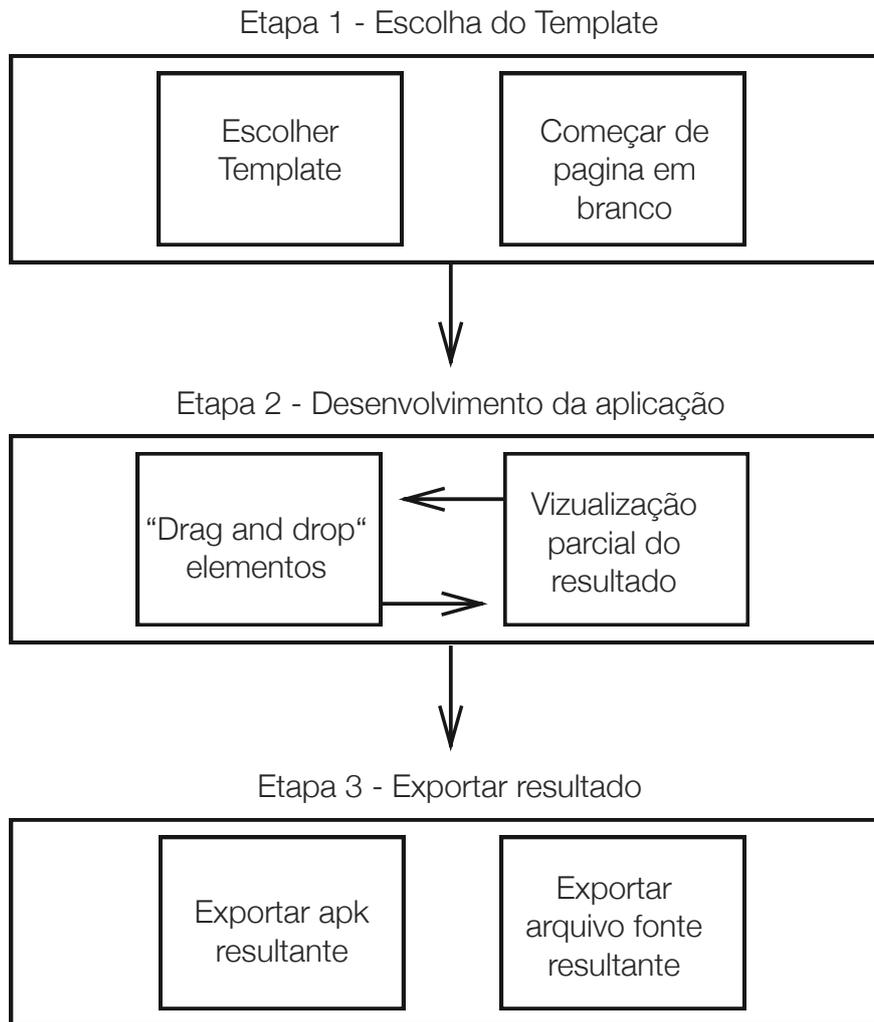
<sup>11</sup> Linguagem de programação interpretada para desenvolvimento web

<sup>12</sup> Extensão de arquivo dos aplicativos iOS

<sup>13</sup> Extensão de arquivo dos aplicativos Android.

<sup>14</sup> Modelo de documento

Figura 5 – Arquitetura do Webflow.



#### Etapa 1 - Escolha do template.

Na figura 2.2, a etapa 1 chamada de "escolha do template" é a primeira fase no desenvolvimento de uma aplicação utilizando a ferramenta *WebFlow*. A ferramenta *WebFlow* como citado anteriormente, utiliza fortemente de elementos de interface e funciona da forma *drag and drop*, porém para iniciar o processo, é solicitada que o usuário selecione uma de 2 (duas) opções apresentadas:

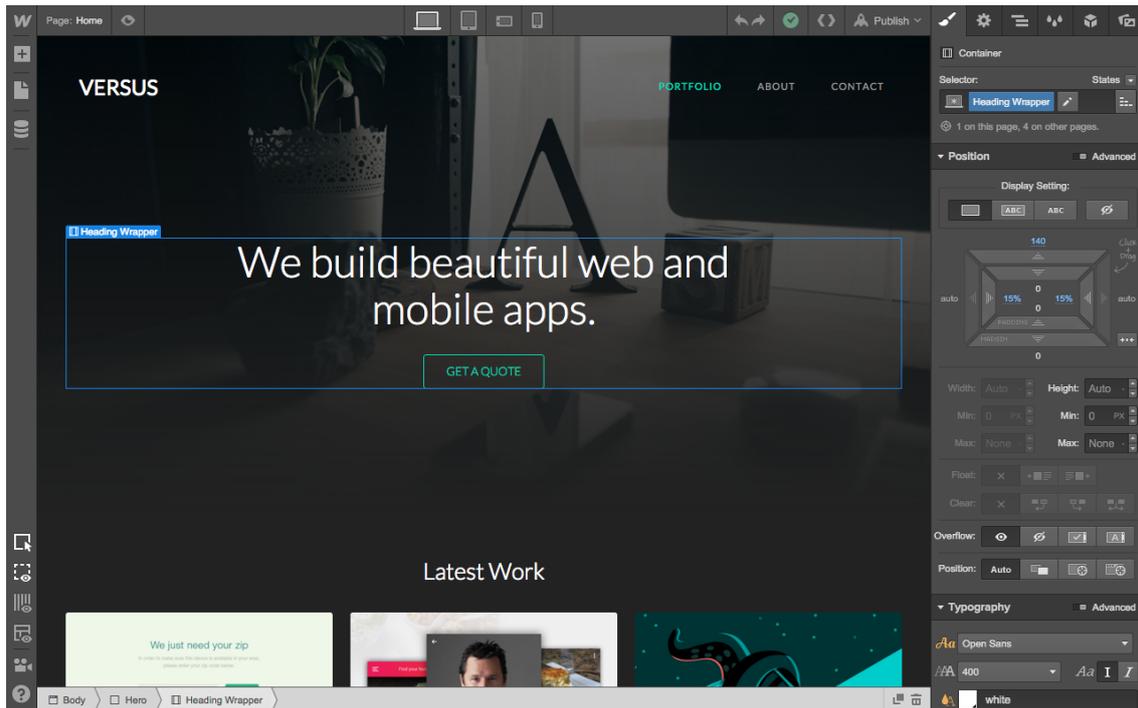
- Escolher um template, como exibido na figura 2.2, na etapa 1.1. feita essa escolha, a ferramenta oferece uma lista de *templates* pré feitos, com toda a estrutura básica, afim de agilizar o desenvolvimento da aplicação. Com o *template* escolhido, o usuário vai ter toda a liberdade do *canvas* para editar os elementos do *template*, assim como arrastar novos elementos para o *canvas*.
- Começar de página em branco, como mostrado na figura 2.2, na etapa 1.2. Feito isso, o que é feito para o usuário é a geração de um *canvas* em branco, para que este possa iniciar um projeto novo. Juntamente como *canvas*, nessa etapa é fornecido para o

usuário todas as ferramentas e elementos disponíveis para que este possa criar sua aplicação.

## Etapa 2 - Desenvolvimento da aplicação.

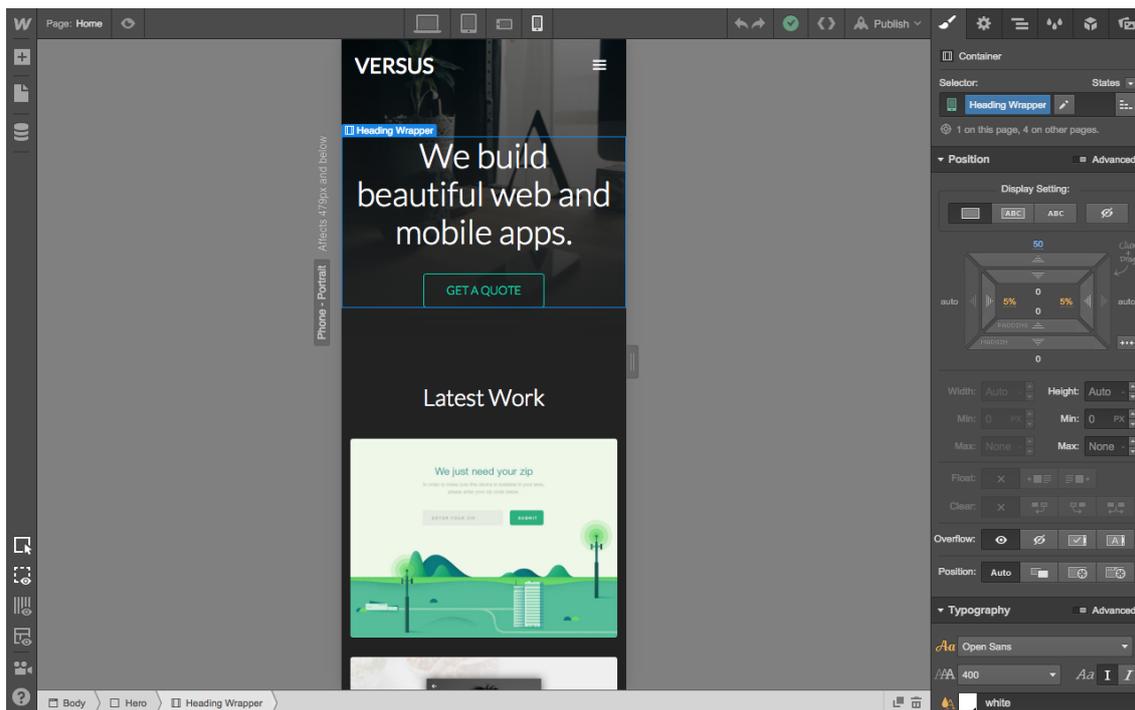
A etapa 2 na figura 2.2, é a etapa onde o usuário dá início ao desenvolvimento da aplicação. Como pode ser visto na figura 2.2, a interface de desenvolvimento da aplicação é toda baseada em configurar os elementos que estão no *canvas*, assim como arrastar novos elementos que prefira. A figura 2.2 é uma tela da interface web do *WebFlow* exemplificando o que é o *canvas* e as ferramentas de desenvolvimento da aplicação.

Figura 6 – Tela da etapa 1 - Template Selecionado



A figura 2.2 mostra de forma mais detalhada o *canvas* e suas ferramentas de edição dos módulos e da aplicação. Como pode ser notado, cada tipo de elemento tem uma área de edição de seus parâmetros, onde pode ser alterado diversos tipos de características, como cores, tamanho do elemento, dentre outros. Na figura 2.2, pode notar que é possível mudar o tipo de exportação para relacionar com cada tipo de aparelho que esteja sendo desenvolvido de forma simples e direta.

Figura 7 – Tela da etapa 2 - Desenvolvimento da aplicação.

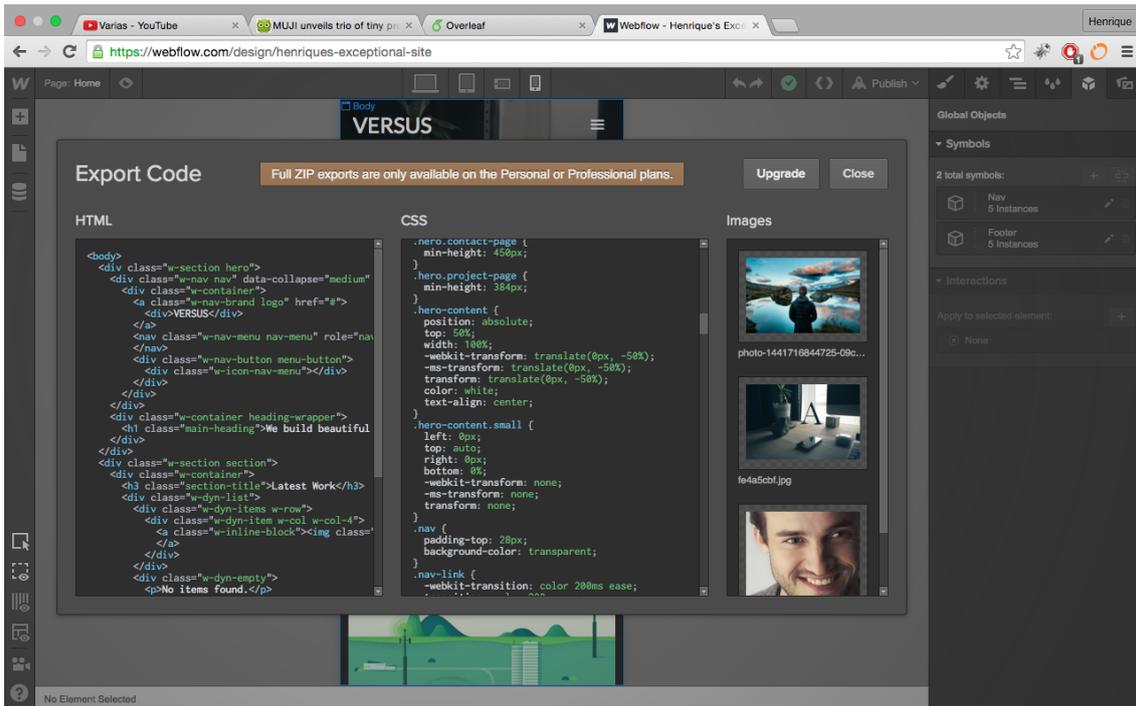


### Etapa 3 - Exportar resultado

A terceira e última etapa, como mostrado na figura 2.2, é a etapa de exportação de resultado. Com o desenvolvimento da aplicação finalizada, o usuário tem 2 opções distintas:

- Exportar *.apk* resultante, com essa escolha selecionada, o que vai ser exportado é um arquivo no formato *.apk*, formado dos aplicativos para a plataforma *Android*. Esse arquivo, pode ser enviado para aprovação la loja de aplicativos da *Google*.
- Exportar arquivo fonte resultante, essa escolha irá gerar e exportar para o usuário, todos os arquivos editaveis da aplicação, imagens que foram adicionadas ao projeto e arquivos de controle. Na imagem 2.2 esta mostrando como é quando o usuário seleciona exportar o código fonte da aplicação.

Figura 8 – Tela da etapa 3 - Exportação do resultado.



## 2.3 AppMachine

O aplicativo *AppMachine* é um software para fazer aplicativos nativos para *iPhone*<sup>15</sup> e *Android*<sup>16</sup> (APPMACHINE, 2013), será utilizado para a representar ferramentas que utilizam de blocos de pseudocódigo para criar a aplicação final. Cada novo bloco utilizado a ferramenta escreve em um arquivo o código relacionado ao bloco e ao final do processo gera a aplicação final. A ferramenta também faz uso de um canvas de desenvolvimento, a principal diferença para a ferramenta *WebFlow* é que no canvas do *AppMachine* o que se utiliza são os blocos de pseudocódigo, enquanto que na ferramenta *WebFlow* é utilizado os elementos gráficos.

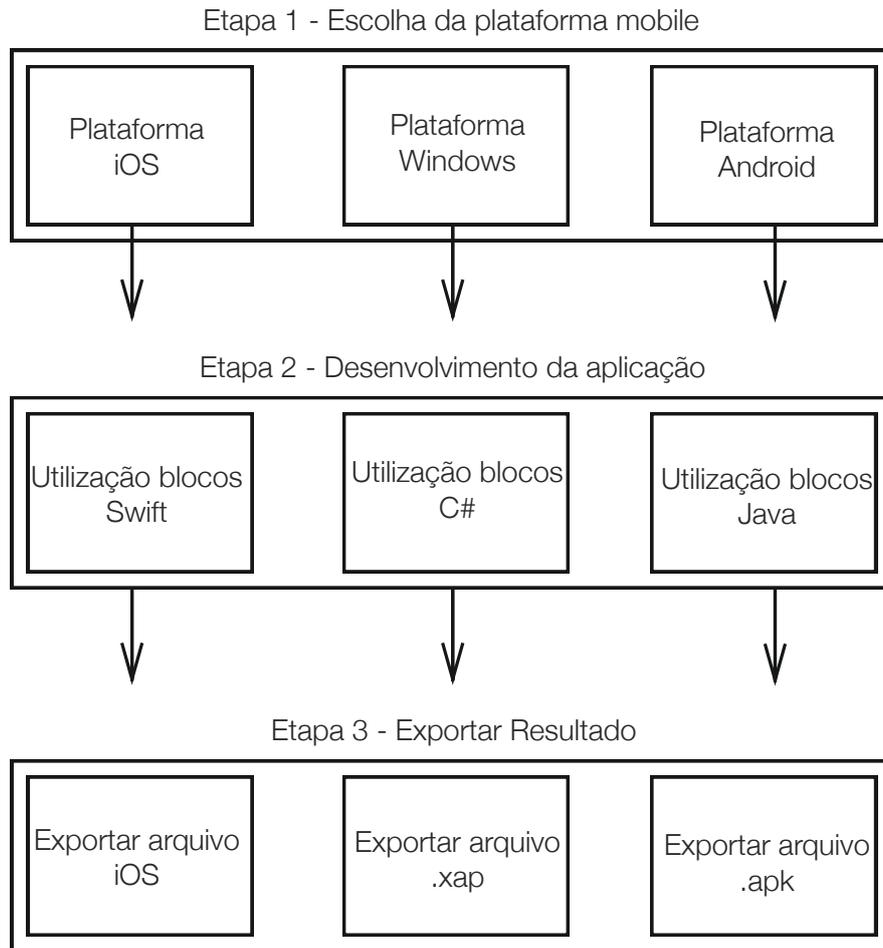
Ao contrario das outras ferramentas que estão sendo utilizadas nesse artigo, a utilização da ferramenta *AppMachine* é um pouco mais complexa. O usuário que faz uso da ferramenta precisa ter conhecimento básico de lógica de programação para que possa construir aplicações mais interativas. Ao final do processo de desenvolvimento, o usuário tem a opção de enviar sua aplicação direto para as respectivas lojas de aplicativos, e com essa opção selecionada, a ferramenta gera o respectivo arquivo dependendo da plataforma através dos arquivos que foram escritos utilizando os blocos de pseudocódigo.

A arquitetura e funcionamento da ferramenta em compensação é a mais simples dentre as 3 (três) ferramentas. A imagem a seguir exemplifica essa arquitetura com as explicações de cada etapa.

<sup>15</sup> Aparelho smartphone da empresa norte americana Apple

<sup>16</sup> Sistema operacional mobile da empresa norte americana Google

Figura 9 – Arquitetura da ferramenta AppMachine.



Etapa 1 - Escolha da plataforma mobile. Diferente das outras ferramentas listadas nesse artigo, a ferramenta *AppMachine* não gera automaticamente a versão do aplicativo para cada uma das plataformas, é necessário que o usuário retorne ao desenvolvimento e faça mudanças no seu canvas que contem os blocos de pseudocódigo. Na imagem 2.3 isso fica claro na etapa 1 - escolha da plataforma mobile. A escolha do usuário de qual plataforma irá desenvolver muda a estrutura de funcionamento das próximas etapas.

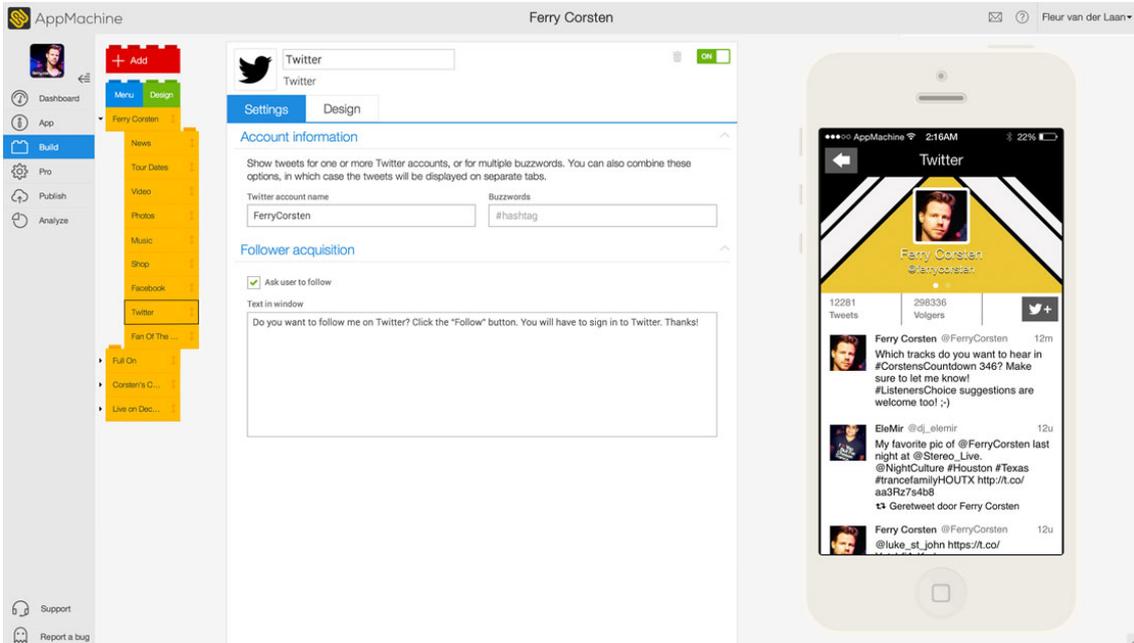
Etapa 2 - Desenvolvimento da aplicação.

Com a escolha da plataforma desejada selecionada, a etapa 2 - desenvolvimento da aplicação na imagem 2.3, a estrutura da ferramenta é modificada para acomodar as ferramentas e opções relativas a plataforma escolhida. Como a ferramenta *AppMachine* faz uso de um canvas, onde o usuário coloca nesse canvas blocos de pseudocódigo afim de representar ações e código nativo, com a plataforma escolhida, os blocos que serão exibidos para serem utilizados são relativos as limitações e características da linguagem da plataforma, podendo existir funcionalidades a mais, estruturas lógicas distintas e maior ou menos flexibilidade no desenvolvimento.

Na etapa 2 - Desenvolvimento da aplicação, o usuário vai construir toda a aplicação que será gerada no final, desde o estilo estetico da aplicação, até suas interações e informa-

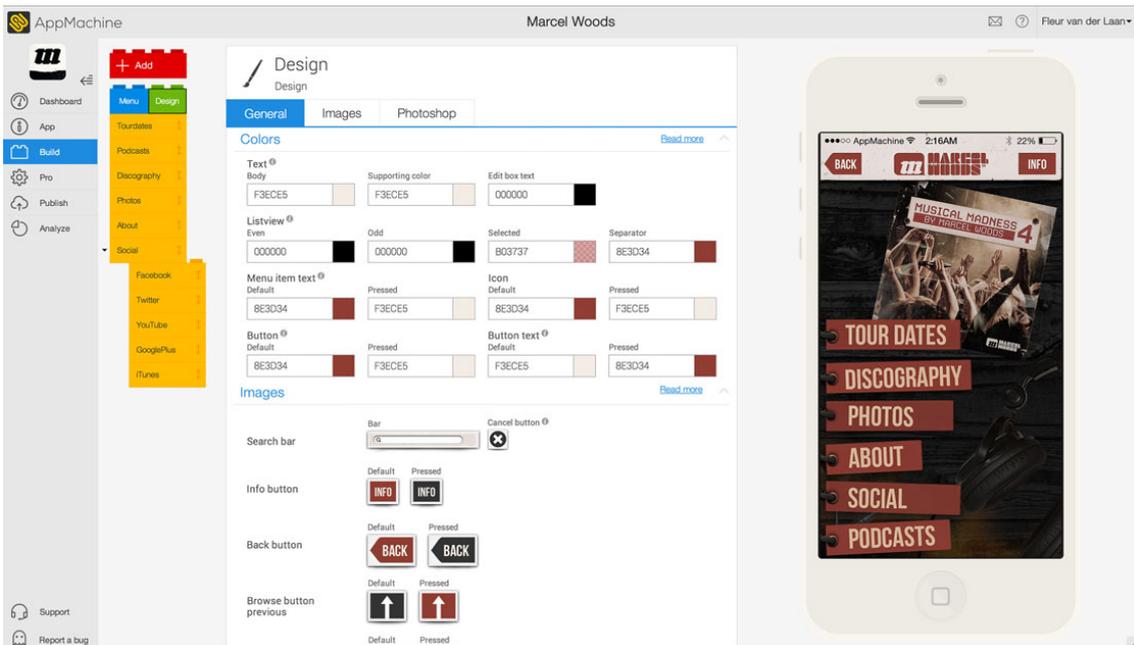
ções técnicas, portanto essa etapa é a principal na utilização da ferramenta. A figura 2.3 é uma tela que exemplifica o desenvolvimento de uma aplicação, mais especificamente, o preenchimento de uma das informações do estabelecimento.

Figura 10 – Preenchimento das informações específicas da aplicação.



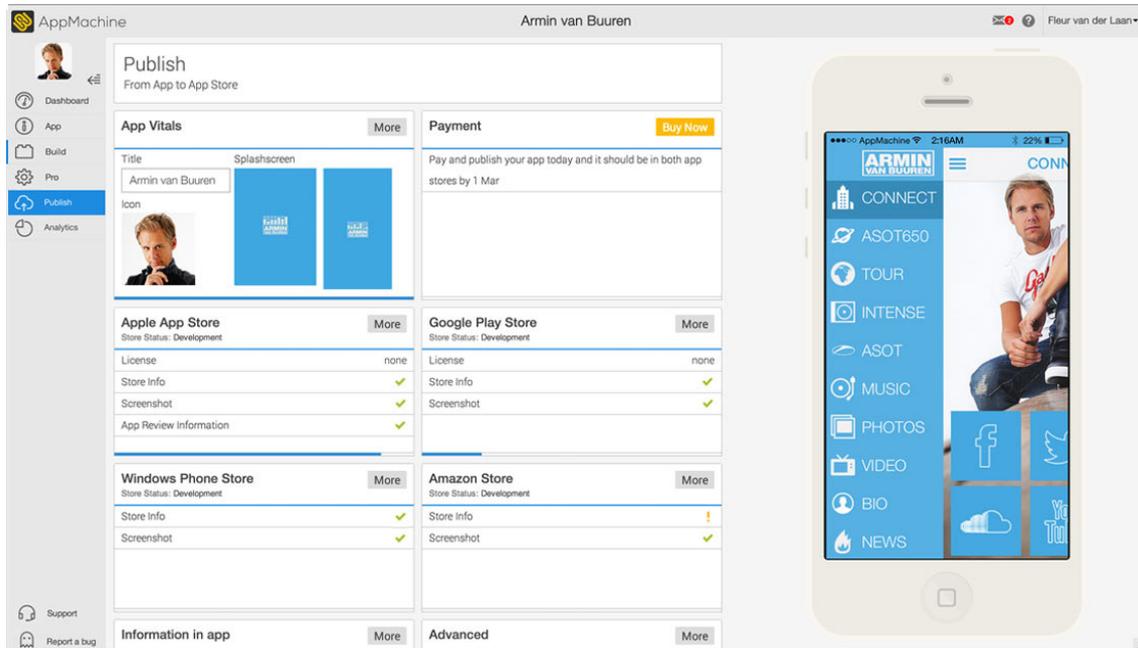
A figura 2.3 mostra outra tela do desenvolvimento de uma aplicação, nesse caso, a escolha dos estilos do aplicativo final. No desenvolvimento da interface do aplicativo, o usuário tem a liberdade de editar alguns aspectos dos elementos, como cores e formato, além de poder adicionar imagens ilustrativas

Figura 11 – Escolha do estilo da aplicação.



Etapa 3 - Exportar Resultado. A terceira e ultima etapa da criação de uma aplicação móvel utilizando a ferramenta *AppMachine*

Figura 12 – Exportar resultado final.



### 3 METODOLOGIA

#### 3.1 Etapas do desenvolvimento

O desenvolvimento desse artigo se deu inicio na criação da ferramenta *AppGenerator*. No inicio do processo foi feita uma busca e comparação com outras ferramentas de desenvolvimento de aplicações móveis afim de encontrar formas de inovar e criar novos métodos, mais faceis e rapidos para gerar aplicações movéis. Durante esse periodo foi escolhido comparar com 2 ferramentas que realizavam o mesmo processo, porém de formas distintas: *WebFlow*, uma ferramenta que já tinha sido muito utilizada em outros trabalhos, e a ferramenta *AppMachine* uma ferramenta que é considerada muito robusta por profissionais da area.

A segunda etapa do desenvolvimento do artigo, iniciada após o termino da criação da ferramenta *AppGenerator* e com ela já em funcionamento, foi a de valida-la no mercado e possíveis clientes e usuários. Durante essa validação, foram realizadas testes de usabilidade com os clientes que já faziam uso da ferramenta. Esse processo teve uma duração de 5 meses, período no qual foram feitos testes exaustivos.

Com um número de testes feitos em uma base razoável de usuários, a terceira etapa foi a de compilar os dados encontrados, para mensurar a qualidade das aplicações. Os dados encontrados, juntamente com o questionário feito de experiencia na utilização das ferramentas, forneceu um quadro claro das vantagens e desvantagens de cada aplicação.

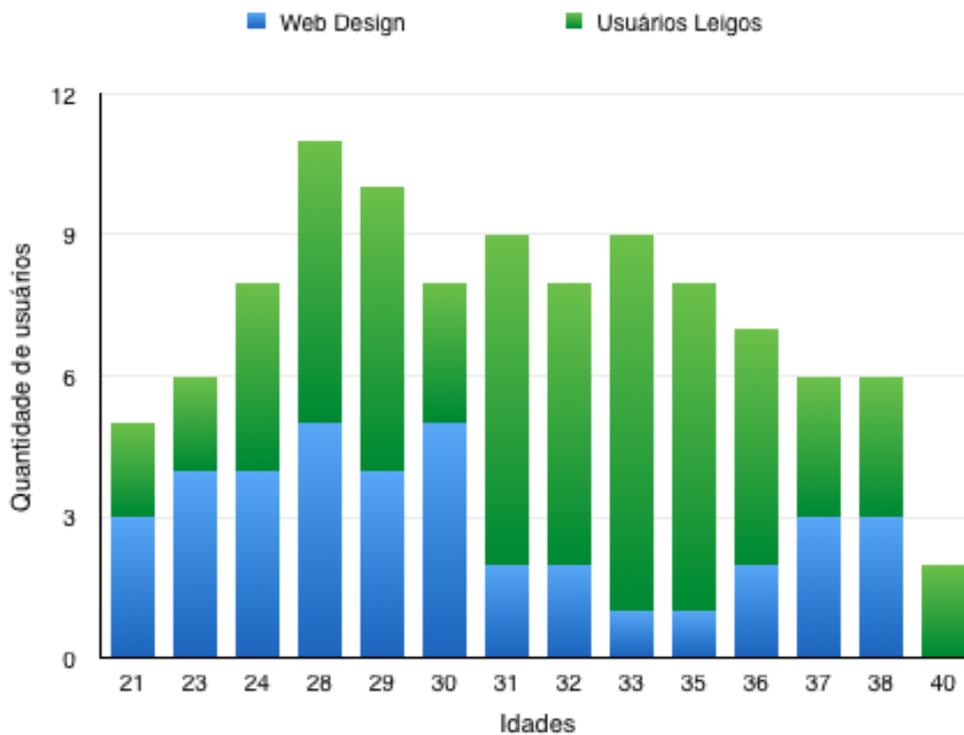
A quarta e ultima etapa, foi dar inicio a documentação do artigo, descrevendo cada uma das ferramentas, juntamente com sua arquitetura e funcionamento, além de mostrar e explicar sua utilização na execução dos testes através de imagens da interface gráfica.

### 3.2 Descrição dos testes

Para conseguir um bom resultado no artigo, foi preciso fazer testes com o maior número de usuários possível, dentre esses usuários, foi preciso ter alguns perfis distintos: Usuários sem conhecimento em programação *hardcore*, ou seja, conhecimento em alguma linguagem de programação, usuários que trabalhem com *web design*, afim de validar as ferramentas em um ambiente de produção.

Com essa necessidade foi utilizado um total de 97 pessoas seguindo o perfil descrito anteriormente. Para realizar os testes, foi utilizado métodos para validar a usabilidade de ferramentas. Como as ferramentas são todas de fácil acesso e disponíveis no mercado, os testes foram feitos abordando grupos de pessoas em empresas de *design* e em locais comerciais afim de validar com usuários leigos em linguagens de programação, de várias faixas etárias.

Figura 13 – Perfil dos usuários teste.



Como podemos observar na figura 3.2, 62 pessoas do total de 97 usuários que foram testados, são pessoas leigas quando se trata de linguagens de programação, enquanto que 39 tem algum tipo de conhecimento prévio.

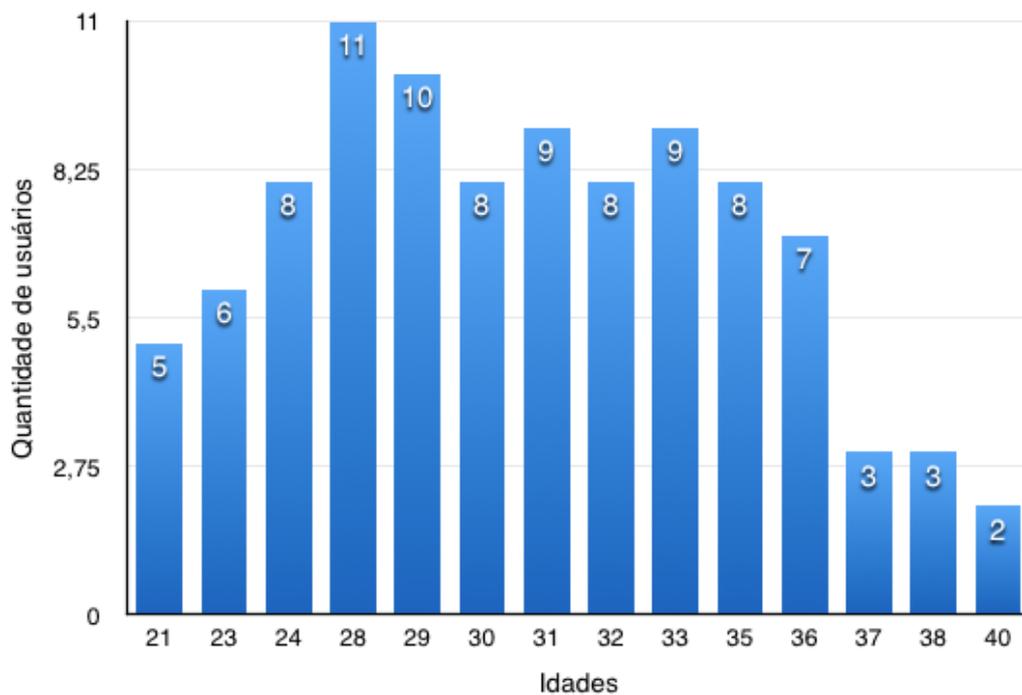
Para realizar o teste foi utilizada a seguinte metodologia: Para cada um dos usuários foi fornecido um roteiro descrevendo o caso de uso que foi elaborado para manter um padrão nos testes, que irá ser melhor especificado na próxima seção, e posicionou o usuário teste em frente a uma câmera para gravar as reações, foi cronometrado cada um deles do início do desenvolvimento da aplicação, e foi utilizado o mesmo computador em todos os testes afim de eliminar a interferência de um hardware diferente nos resultados.

Em conjunto com o roteiro que precisava ser executado para a realização dos

testes, foi fornecida ao final da avaliação um questionário para cada usuário, para que este pudesse relatar sua experiência e dificuldade na utilização das ferramentas para realizar os testes. Esse questionário, que vai ser melhor explicado na subseção seguinte, tem como objetivo final entender qual a melhor ferramenta para cada grupo distinto de usuários.

Os testes foram executados ao longo de um período de 5 meses, onde um total de 97 pessoas foram avaliadas utilizando as 3 (três) ferramentas, com idade variando entre 21 e 40 anos. Durante esse período foi possível perceber um padrão na utilização das ferramentas por parte dos usuários. Os usuários mais novos e com maior acesso a tecnologia, tiveram maior facilidade utilizando da ferramenta *WebFlow*, enquanto os usuários de idade mais avançada, se sentiram mais confortáveis em utilizar a ferramenta *AppGenerator*. Na imagem 3.2 é possível notar que dentre os 97 usuários teste, a grande maioria está na faixa etária de 24 e 35 anos.

Figura 14 – Quantidade e idade dos usuários testes.



Com os testes feitos e cronometrados, foi realizada uma tabulação dos resultados para melhor entender os padrões de utilização. Foi feita uma média dos valores nas 3 medidas de avaliação, tempo de desenvolvimento, número de linhas gerado e o tamanho do aplicativo resultante, com o desvio padrão de cada uma delas. Os resultados completos com a explicação será melhor descrito na seção de avaliação.

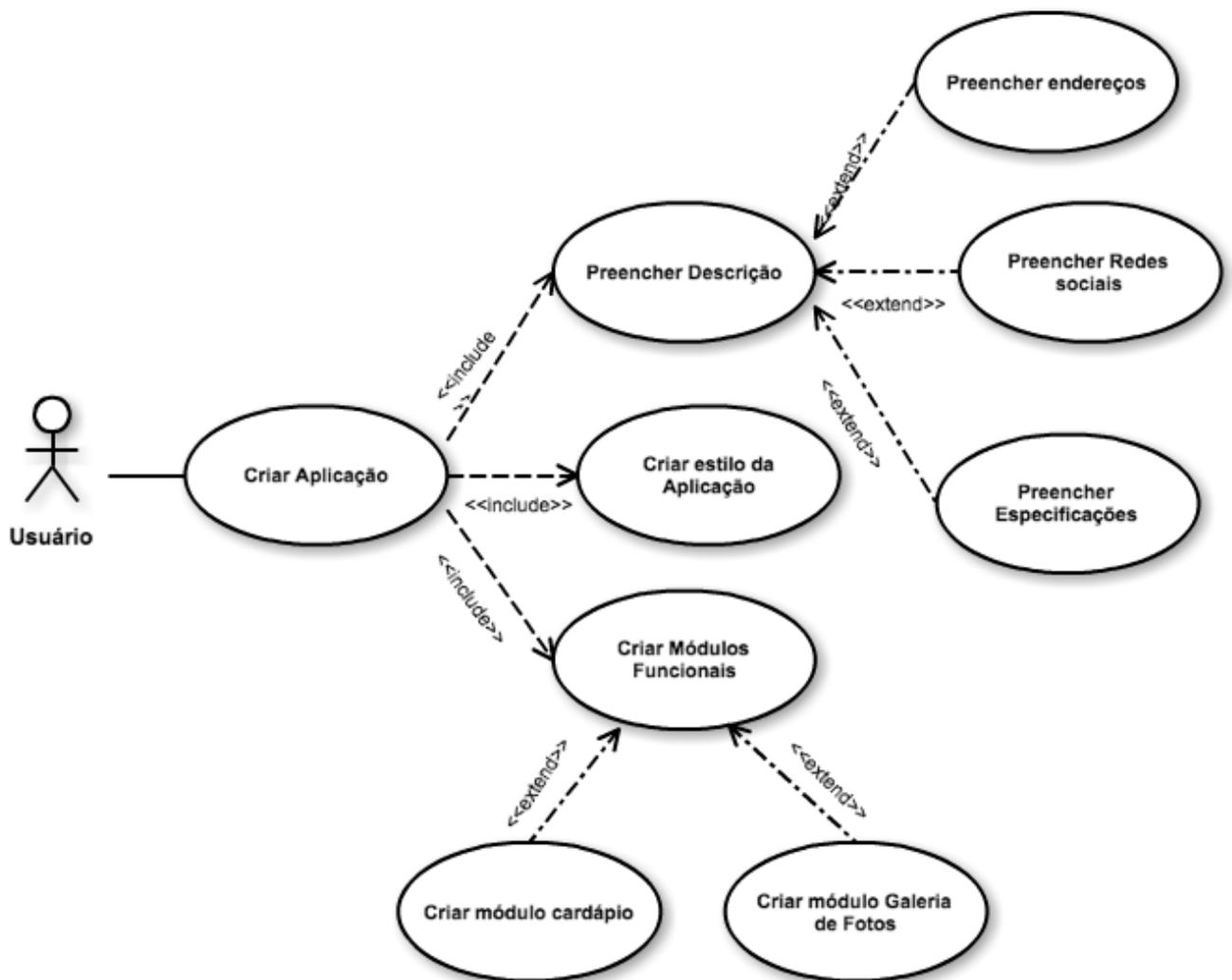
### 3.3 Descrição da aplicação teste

Para comparação das ferramentas foi escolhida um caso de uso real: A criação de uma aplicação para um estabelecimento comercial. O objetivo dessa aplicação é conter informações básicas do estabelecimento, como horário de funcionamento, história, o cardápio, comentários, e galeria de fotos.

No final do processo será comparada os resultados, e gráficos comparativos. O processo de comparação será feito seguindo 2 (duas) etapas, a primeira, a criação da aplicação em cada uma das ferramentas. E a segunda etapa, será feito um comparativo do resultado final nas métricas de: numero de linhas geradas, o tamanho em bytes do produto gerado e a velocidade média no desenvolvimento utilizando as 3 (três) ferramentas.

O processo de desenvolvimento da aplicação foi feita utilizando uma lista de requisitos necessários na aplicação final, cujo objetivo é padronizar o desenvolvimento. Segue o diagrama de caso de uso da aplicação teste necessária. Na figura 3.3 é possível notar os elementos que o usuário precisa desenvolver com as 3 (três) ferramentas. É importante enfatizar que é possível realizar essa tarefa com as ferramentas escolhidas para esse artigo

Figura 15 – Questionário de avaliação de Avaliação de Experiência.



A figura 3.3 exibe o que será preciso ser feito. A descrição a seguir, mostra os campos, estilo e estrutura de cada um desses elementos. Essa escolha é para padronizar o teste em relação a estrutura, e que os elementos tenham o mesmo estilo e campos, afim de mostrar a dificuldade de se criar a aplicação teste em cada uma das ferramentas.

### 1. Etapa de descrição.

- Preenchimento do endereço completo: País, bairro, rua, número, descrição.
- Preenchimento das redes sociais: *facebook*, *twitter*, telefone, web site.
- Preenchimento das especificações: Horário de funcionamento, formas de pagamento, ambiente, política de animais, cardápio especial, tipo de comida, tipos de restaurantes, recomendação, infraestrutura de deficientes, especificação de estacionamento.

### 2. Etapa de descrição do estilo

- Cor do menu: Branco.
- Cor do cabeçalho: Azul.
- Cor do conteúdo: Preto.
- Estilo do menu: Estilo 1

### 3. Preenchimento dos módulos

- Módulo do cardápio: 1 seção, 2 subseções, 4 pratos.
- Módulo de galeria de foto: 1 galeria, 10 fotos.

## 4 RESULTADOS

Após feitos os testes, vários resultados foram encontrados. Para os testes foram utilizados uma base de usuários de 97 pessoas, dentre elas *web designers* e pessoas sem conhecimento em desenvolvimento de software e linguagens de programação. Os testes foram cronometrados em cada uma das etapas. A seguir segue 1 (uma) tabela contendo os resultados encontrados e a explicação do que foi observado. A tabela esta dividida entre as 3 (três) ferramentas que foram comparadas, e cada linha contém o resultados dos testes feitos: número médio de linhas total da aplicação, tempo médio de desenvolvimento da aplicação e o tamanho em bytes médio da aplicação.

Figura 16 – Tabela de resultados.

	AppGenerator		Wordpress		AppMachine	
	Media	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
Tempo de desenvolvimento	22m	7m	1h 10m	19m	1h 56m	23m
Número de Linhas	5776	941	13693	1789	21179	4409
Tamanho do App (MB)	5,056	0,886	--	--	5,897	0,465

O tempo de desenvolvimento na figura 4 é o primeiro fator analisado e o de maior índice de discrepância entre os 3 (três) testes. Para fazer essa análise, foi cronometrado a utilização dos 3 (três) sistemas enquanto se desenvolvia a aplicação alvo. Como resultados foi constatado que com a ferramenta *AppGenerator* é muito mais rápido desenvolver

uma aplicação web e aplicação *mobile*, a média no tempo de desenvolvimento utilizando a ferramenta *AppGenerator* foi de 22 minutos com um desvio padrão de 7 minutos, aproximadamente 3 (três) vezes mais rápida que utilizando a ferramenta *WebFlow* e 5 (cinco) vezes mais rápida que utilizando a ferramenta *AppMachine*, enquanto utilizando a ferramenta *WebFlow* o tempo médio foi de 1 hora e 10 minutos, com um desvio padrão de 19 minutos, e por ultimo, utilizando a ferramenta *AppMachine* o tempo de desenvolvimento médio foi de 1 hora e 56 minutos, com um desvio padrão de 23 minutos. Isso se deve por alguns motivos, o primeiro é o fato de que com a ferramenta *AppGenerator* é possível desenvolver a aplicação web e a aplicação *mobile* simultaneamente, enquanto que na ferramenta *WebFlow* só é possível o desenvolvimento da aplicação web, e na ferramenta *AppMachine* só é possível desenvolver aplicação *mobile*. Outro aspecto crucial é o fato de que a ferramenta *AppGenerator* o desenvolvimento já tem uma base predeterminada, enquanto nas outras ferramentas o usuário precisa escolher e desenvolver cada uma das funcionalidades desejadas.

Na figura 4 o segundo resultado constatado foi o número de linhas da aplicação gerada. Essa análise foi feita separadamente entre as ferramentas devido suas limitações: a ferramenta *WebFlow* não gera o aplicativo *mobile*, e a ferramenta *AppMachine* não gera o aplicativo web. A primeira comparação feita foi a do número de linhas da aplicação web gerada pela ferramenta *AppGenerator* e a aplicação web gerada pela ferramenta *WebFlow*, onde o número de linhas total, contando de todos os diferentes arquivos foi inferior na ferramenta *AppGenerator* em quase 3 (três) vezes, com um número total médio de 5776 linhas em comparação com a ferramenta *WebFlow* com um número total médio de 13693. A segunda comparação feita foi a do número de linhas da aplicação *mobile* gerada pela ferramenta *AppGenerator* e a ferramenta *AppMachine*, onde nesse caso, o número de linhas total, também contando as linhas de todos os arquivos foi inferior na ferramenta *AppGenerator* em 3,2 vezes, com uma média de 6467 linhas, enquanto na ferramenta *AppMachine* a média total foi de 21179 e desvio padrão de 4409 linhas.

A conclusão que se chega para esse número inferior, é devido a como a ferramenta *AppGenerator* se diferencia das outras, ela tem uma base de arquivos feita de forma muito otimizada e somente o estilo e módulos escolhidos são preenchidos, ao contrário das outras 2 (duas) ferramentas que usam *Snippets*, pequenas regiões de código reutilizável, para gerar o código final, dessa forma o código gerado é de qualidade inferior, contendo muita informação desnecessária. Outra conclusão que se chega é a alta superioridade na manutenção da aplicação gerada pelo *AppGenerator*, isso se deve ao número inferior de linhas de código e maior organização do resultado gerado, o que facilita a manutenção.

O terceiro e ultimo resultado constatado exibido na 4 foi o tamanho em *MegaBytes* de cada aplicação gerada. Essa análise foi feita como a avaliação do número de linhas, separando as ferramentas devido suas limitações. O que pode ser constatado em ambos os casos é que o número de linhas é relativamente similar nas 3 (três) aplicações. O motivo disso é o fato de que como a aplicação desenvolvida não é muito complexa, contendo basicamente texto e imagens, o tamanho em *MegaBytes* é relativo ao numero de linhas e arquivos gerados, e esse número por mais discrepante que seja, não faz o tamanho do arquivo final gerado diferir em muito. Na aplicação gerada pela ferramenta *AppGenerator* o tamanho médio da aplicação *mobile* foi de 5,05 MB <sup>17</sup>, enquanto na ferramenta *AppMachine* o tamanho médio foi de 5,89 MB.

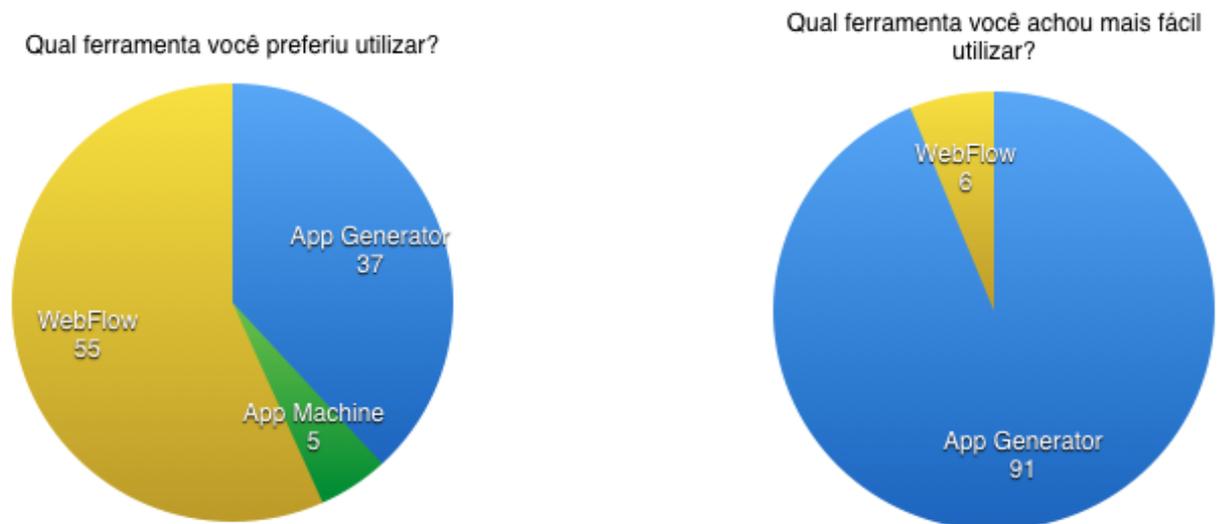
Juntamente com os testes que foram feitos, foi fornecido aos usuários um ques-

<sup>17</sup> Unidade de medida de informação que equivale a 1 000 000 bytes

tionário para melhor entender sua relação e experiência durante o uso das ferramentas afim de entender suas dificuldade e impressão. O que o questionário mostrou foi que os usuários mais jovens, entre 21 e 27 anos, preferiram a ferramenta *WebFlow*, com uma taxa de preferência de 57 por cento ou 55 usuários, a conclusão encontrada foi que esses usuários preferem ter uma maior liberdade na hora de elaborar uma aplicação, sem precisar de conhecimentos em linguagens de programação que o desenvolvimento *hardcore* necessita. A ferramenta *WebFlow* permite uma maior flexibilidade no uso dos parâmetros o que torna sua utilização mais robusta. Em contraste com esse resultado, foi percebido que os usuários com idade mais avançada, tiveram maior facilidade em utilizar a ferramenta *AppGenerator* por ser uma ferramenta que fornece uma utilização mais didática, onde em cada etapa distinta do processo é oferecida uma quantidade limitada de escolhas, e ao final do processo tem como resultado a aplicação. A ferramenta *AppMachine* teve um resultado unânime que sua utilização é a mais trabalhosa, e a que exige um maior conhecimento em linguagens de programação e em lógica de programação, por fazer uso de blocos que representativos de pseudocódigo. Como os usuários teste não tinham conhecimento em linguagens de programação, a sua utilização foi confusa e a curva de aprendizado muito superior quando comparada com as outras 2 (duas) ferramentas, com uma preferência de apenas 5 por cento ou 5 usuários.

Na figura 4 é possível notar a ferramenta que teve a melhor recepção na utilização foi a *WebFlow*, esse valor tem relação pela grande quantidade de usuários que trabalham com *Web Design*, devido ao fato que esta ferramenta proporciona uma maior liberdade nas mudanças dos parâmetros de cada elementos. Em compensação é possível comprovar que a ferramenta que os usuários acharam mais fácil de se utilizar foi a *AppGenerator*, com uma quantidade esmagadora de 91 usuários do total de 97.

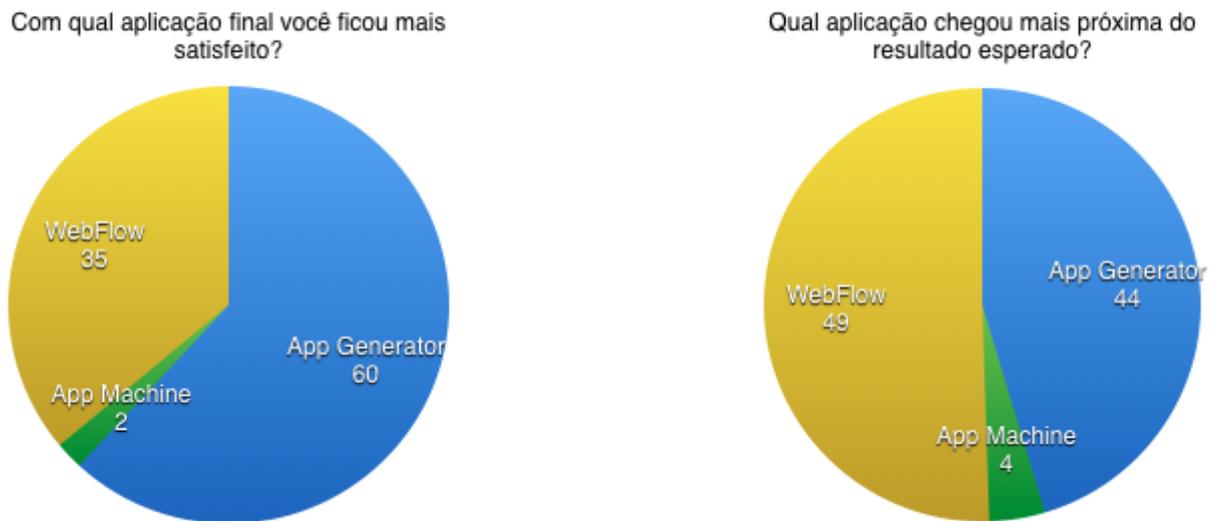
Figura 17 – Gráfico contendo as respostas do questionário de avaliação da Experiência.



Como mostrando na figura 4 os resultados encontrados pelo questionário de avaliação de experiência, mostrou que a aplicação resultante da ferramenta *AppGenerator* e da ferramenta *WebFlow* foram as aplicações que mais agradaram e chegaram próximas do resultado esperado. Enquanto que a ferramenta *AppMachine*, devido a sua maior dificuldade em utilizar, teve o pior resultado das ferramentas. Os numero de usuários que aprovaram

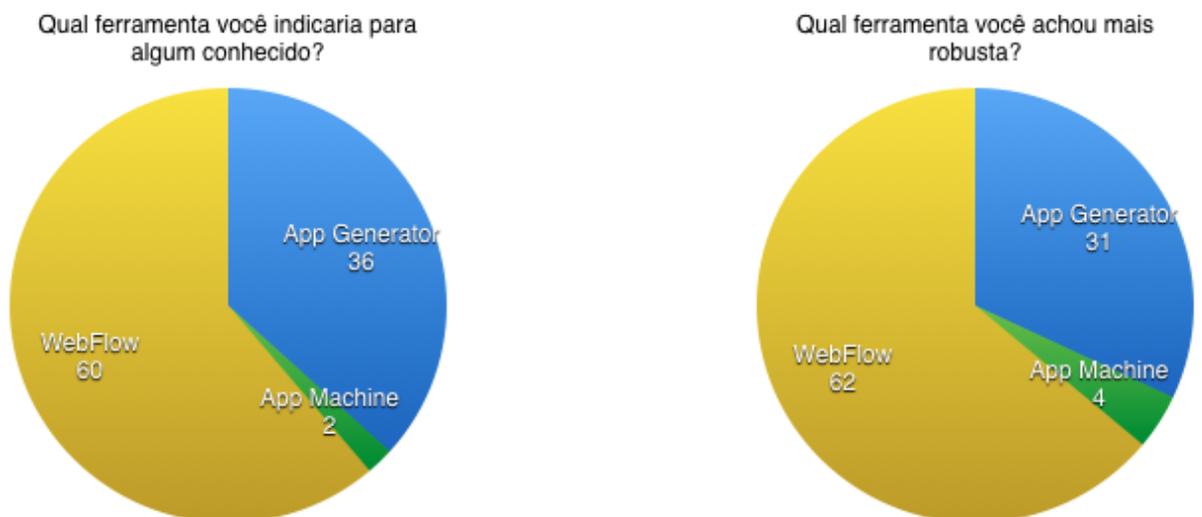
a aplicação gerada pela ferramenta *App Generator* foi de 60 usuários, e na ferramenta *WebFlow* foi de 35 pessoas.

Figura 18 – Gráfico contendo as respostas do questionário de avaliação da Experiência.



A terceira imagem que mostra os resultados do questionário feito com os usuários comprova a superioridade em quesitos de usabilidade da ferramenta *WebFlow*, que se deve ao fato dela permitir uma maior flexibilidade na mudanças do estilo e parâmetros dos elementos, característica muito desejada pelos *web Designers*.

Figura 19 – Gráfico contendo as respostas do questionário de avaliação da Experiência.



## 5 CONCLUSÕES

Nesse artigo foi apresentado 3 (três) ferramentas de desenvolvimento de aplicações WEB e aplicativos Mobile, que oferecem uma maneira mais rápida e eficiente de gerar aplicações web e aplicativos *mobile*. Foi proposto um caso de uso, onde uma base de usuários de 97 pessoas foi solicitado a desenvolver uma aplicação real de um estabelecimento e ao final do processo foi reunido os resultados.

Foi apresentado um estudo de caso, demonstrando o poder e facilidade da ferramenta, e um exemplo de resultados. As comparações foram feitas de 3 (três) formas distintas, a primeira, a velocidade no desenvolvimento da aplicação mostrou resultados mais rápidos na ferramenta *AppGenerator* em um fator de 3 (três) vezes quando comparado a ferramenta *WebFlow* e 5 (cinco) vezes quando comparado a ferramenta *AppMachine*. A segunda maneira comparada foi a de número total de linhas na aplicação gerada, que mostrou um número inferior na aplicação *AppGenerator* quando comparado com as outras 2 (duas) ferramentas em 2,3 vezes, com isso é possível concluir que o código gerado pela ferramenta *AppGenerator* é superior em qualidade e com uma maior reusabilidade e manutenção.

A segunda parte de validação foi feita utilizando um questionário para validar o uso das ferramentas. E os resultados mostraram que juntamente com os quesitos objetivos mostrados nos testes, o quesito de experiência do usuário é tão importante para o usuário final, quanto a qualidade do resultado. O questionário mostrou que a ferramenta *WebFlow* fornece uma experiência superior para a maioria dos usuários, em 56 por cento superior quando comparado com a ferramenta *AppGenerator* por fornecer uma flexibilidade maior que é muito desejada pelos profissionais da área de *web design*.

A principal dificuldade na execução do artigo, foi conseguir um numero significativo de usuários, de várias faixas etárias e de conhecimentos distintos para testar as 3 ferramentas. A necessidade de se utilizar um grande número de usuários é importante para validar a usabilidade e qualidade da aplicação gerada em diferentes grupos. O trabalho apresentado apresentou 3 métricas objetivas para comprar a qualidade da aplicação gerada: velocidade no desenvolvimento da aplicação, número total de linhas na aplicação gerada e o tamanho em bytes da aplicação gerada. O trabalho também apresentou uma métrica subjetiva, que vem da opinião dos usuários testes quando estes utilizaram o sistema. Esse numero de métricas foi limitado por se tratar de uma grande quantidade de usuários que iriam realizar os testes. A expectativa é que nos trabalhos futuros seja feito uma comparação com novas métricas e um numero maior de ferramentas que geram aplicações.

Atualmente a ferramenta *AppGenerator* tem recebido aproximadamente 26.000 acessos mensais de web *designers* e usuários interessados na criação de aplicações web. A tendência do mercado mostra que a perspectiva é que esse número aumente consideravelmente nos próximos anos, e com o advento de novas tecnologias e funcionalidades a ferramenta.

A ferramenta *WebFlow* tem crescido exponencialmente entre o meio de designer gráfico, por proporcionar uma maior flexibilidade no desenvolvimento de Web Sites, onde esta permite codificar através de uma interface gráfica alguns módulos que ainda não existam. A tendência do mercado aponta para um maior uso desse tipo de ferramentas por ser um equilíbrio entre ferramentas *drag and drop* e codificação *harcore*.

A ferramenta *AppMachine* tem se mostrado antiquada quando comparada aos novos sistemas que vem sendo desenvolvimento e mais ainda quando comparada as outras

ferramentas mostradas nesse artigo. A ferramenta que é uma aplicação *desktop* é muito pesada e não tem suporte no principal sistema operacional atual, o *Windows*, da empresa Microsoft. A falta do suporte limita muito a adoção da ferramenta no meio de *designers* e desenvolvedores de sites interativos.

## 6 Apendice

### 6.1 Questionário de avaliação da experiência

Em conjunto com o caso de uso para realizar os testes, foi elaborado um questionário que vai ser aplicado ao final da avaliação afim de colher as impressões dos usuários após os testes e validar a experiência com cada uma das ferramentas. o objetivo desse questionário é testar a experiência e fazer um relacionamento direto com o resultados dos testes de usabilidade.

O questionário foi desenvolvido com um numero total de 6 questões que justas irão fornecer uma visão geral de como foi a utilização das ferramentas. Com o questionário respondido e os testes feitos, vai ser feito um cruzamento de ambos os resultados para comparar os 2 (dois) critérios de qualidade: A qualidade dos sistemas e a qualidade proporcionada pela experiência de uso.

A seguir segue o questionário com as questões que foram utilizadas.

Figura 20 – Questionário de avaliação de Avaliação de Experiência.

## Questionário de Avaliação da Experiência

### Qual ferramenta você preferiu utilizar?

- AppGenerator
- App Machine
- WebFlow

### Qual ferramenta você achou mais facil utilizar?

- AppGenerator
- AppMachine
- WebFlow

### Com qual aplicação final você ficou mais satisfeito?

- Gerada pelo AppGenerator
- Gerada pelo App Machine
- Gerada pelo WebFlow

### Qual aplicação chegou mais próxima do resultado esperado?

- Gerada pelo AppGenerator
- Gerada pelo App Machine
- Gerada pelo WebFlow

### Qual você indicaria para algum conhecido?

- AppGenerator
- App Machine
- WebFlow

### Qual ferramenta você achou mais robusta?

- AppGenerator
- App Machine
- WebFlow

## Referências

APPMACHINE. *appmachine*. 2013. Site do sistema. Disponível em: <[http:](http://)

[//www.appmachine.com/pt-pt/](http://www.appmachine.com/pt-pt/)>. Acesso em: 02 jun. 2015. Citado 2 vezes nas páginas 3 e 13.

CRUNCH, T. *Why you should have your own App*. São Francisco, 2015. Acesso em: 13 abr 2015. Citado na página 2.

EMPRESAS, M. *A importância do mobile para sua empresa*. São Paulo, 2015. Acesso em: 13 abr 2015. Citado na página 2.

GLOBO. *Número de smartphones cresce no Brasil*. São Paulo, 2015. Acesso em: 21 mar 2015. Citado na página 1.

IBGE. *Número de smartphones cresce no Brasil*. Belo Horizonte: Centro de Documentação e Disseminação de Informações. Fundação Instituto Brasileiro de Geografia e Estatística, 2015. Acesso em: 21 mar 2015. Citado na página 1.

VERGE. *Everyone can build now*. São Francisco, 2014. Acesso em: 21 set 2014. Citado na página 2.