

Desenvolvimento de uma biblioteca de ícones para a programação Web

Michel Robaskiewicz Ferreira da Silva¹, Rafael Henrique Dalegrave Zottesso¹

¹Instituto Federal do Paraná (IFPR)
Paranavaí – PR – Brazil

michelrobaskiewicz@gmail.com, rafael.zottesso@ifpr.edu.br

Abstract. *This article aims to develop an icon library for use in Web applications. Its main role is to provide its own icons, such as Web icons, payment methods, technologies, climates, accessibility and vehicles, as well as a series of features such as colors, sizes, effects and rotations to optimize project development. In total, this library offers 300 different icons.*

Resumo. *Este artigo tem como proposta o desenvolvimento de uma biblioteca de ícones para ser usada em aplicações Web. Tendo como papel principal disponibilizar ícones próprios, sendo eles ícones da Web, meios de pagamento, tecnologias, climas, acessibilidade e veículos além de uma série de funcionalidades como: cores, tamanhos, efeitos e rotações para otimizar o desenvolvimento em projetos. No total, esta biblioteca oferece 300 ícones diferentes.*

1. Introdução

As pessoas de hoje estão tão conectadas no mundo digital que é comum ter contato com algum tipo de software durante o dia todo, seja a trabalho, em contato com parentes e amigos, diversão ou qualquer outro tipo de entretenimento. Sempre tem uma solução computacional que faz aquilo que queremos. Essas soluções possibilitam com que o usuário tenha uma interação com suas telas e funcionalidades, de tal maneira que se pode tornar um convívio contínuo. A projeção de telas mais simples por conta da grande diversidade de dispositivos, sejam eles *mobile*, *tablets* ou computadores pessoais, ajudam na navegabilidade e acesso a informações das páginas. Assim, se tornou comum a utilização de figuras pequenas ou ícones que representam ações a serem tomadas. Um bom exemplo disso é quando se anexa um arquivo através do e-mail, nota-se que o ícone é representado por um “*clip*”.

Ao passar do tempo, a frequente utilização permite a memorização das ações que são oferecidas pelos sistemas computacionais por meio da navegabilidade. A utilização de desenhos com o objetivo de representar essas ações existe desde a pré-história, onde os indivíduos que vivenciaram naquele período desenhavam as situações que presenciavam nas paredes das cavernas, desenhos que foram nominados de Arte-rupestre e até hoje ao serem vistos podem ser compreendidos.

Segundo [Marques 2016] a Arte-rupestre é uma das exibições mais antigas criadas pelo homem, tendo objetivo de demonstrar o cotidiano e cultura do homem pré-histórico por meio de desenhos e pinturas em paredes de cavernas ou em esculturas.

Os ícones propostos por esta biblioteca têm o papel de influenciar as ações que podemos tomar em uma solução computacional. Podem ser por meio da navegabilidade entre as páginas, links, funcionalidades e outros modos. Assim, o entendimento por parte dos usuários fica mais claro e intuitivo, pois ao observar um ícone é possível ter uma ideia do que ele está prestes a fazer.

O objetivo deste trabalho é propor a criação de uma biblioteca de ícones que seja funcional, simples e atraente: funcional por ser muito útil no desenvolvimento de páginas Web; simples por ser necessário apenas fazer download, importar as bibliotecas e utilizar as determinadas classes para cada ícone; atraente devido a qualidade e diversidade dos ícones, sendo que muitos deles representam situações e soluções nativas do Brasil. Além disso, destaca-se que esta biblioteca será gratuita.

1.1. Trabalhos Relacionados

As bibliotecas de ícones existentes fornecem pouca opção de ícones que são característicos de coisas do Brasil. A maioria deles condizem com as necessidades presenciadas durante o desenvolvimento de um software Web ou fornecem poucas opções de personalização.

A biblioteca *Font Awesome*¹ na sua versão 4.7 disponibiliza uma grande quantidade de ícones que podem ser baixados gratuitamente em um pacote de arquivo de Folha de Estilo em Cascatas (CSS), tendo como um de seus pontos negativos a disponibilidade apenas de ícones internacionalmente conhecidos. Após um período saiu a versão 5 que disponibiliza uma quantidade limitada de ícones na versão gratuita, pois tem um foco na comercialização da ferramenta.

O *Bootstrap*², tratando-se de ser um *framework* gratuito, em sua 3ª versão disponibilizava apenas uma quantidade limitada de símbolos que pertenciam a uma outra ferramenta chamada *Glyphicons*³, mas que estava embutida internamente em seu funcionamento e podiam ser aplicados durante o desenvolvimento. Na versão 4, que é a atual, descontinuou a disponibilização dos ícones em seu projeto dando espaço para outras bibliotecas de ícones externos serem utilizados. Assim como o *Font Awesome*, o *Glyphicons* também está comercializando pacotes de ícones.

2. Metodologia

O principal propósito desse projeto é o desenvolvimento de uma biblioteca de ícones que forneça uma variedade de ícones funcionais, cores, ações e tamanhos por meio do CSS. A construção do projeto exigiu a utilização de software de desenho vetorial *Adobe Illustrator CS6* para desenhar os ícones, o editor de fonte *FontForge* para criação e manipulação da fonte que será incorporada no CSS e do Ambiente de Desenvolvimento Integrado (IDE) *Sublime text 3* para a codificação. O CSS será o arquivo principal desse projeto, e é a partir dele que as classes serão utilizadas para representar os ícones, tamanhos, cores e efeitos.

¹<https://fontawesome.com/>

²<https://getbootstrap.com>

³<https://www.glyphicons.com/>

2.1. Usabilidade

Os ícones planejados e traçados possibilitam a compreensão dos usuários nas ações que podem representar cada desenho, desse modo possibilita a facilidade de aprendizado e recordação nos sistemas computacionais em que a biblioteca estiver introduzida. A usabilidade está ligada diretamente a essa ferramenta por permitir uma simples interação com os usuários através dos ícones.

Segundo [Nielsen 1993] a usabilidade tem o objetivo de produzir interfaces agradáveis de fácil interação. Ainda enfatiza que a usabilidade pode ser dada em alguns fatores básicos, dentre eles podemos destacar dois, dos quais possibilitam os usuários a terem facilidade no aprendizado e facilidade de recordação.

A interação dos usuários através de telas intuitivas em um produto computacional aproxima e evidencia a importância dos dados e informações serem coerentes, além de permitir os usuários aprendam consigo próprio a navegar.

2.2. CSS - Cascading Style Sheets

A linguagem de marcação *Cascading Style Sheets* mais conhecida como *CSS*, se trata de uma folha de estilo, que proporciona com que páginas *HTML* incorporem suas características e renderize seus estilos definidos, os dois andam juntos. Desse modo proporciona uma excelente experiência ao desenvolvedor e usuários.

[Miyagusku 2007] enfatiza que uma das principais funções da folha de estilo *CSS* é realizar a formatação separando o código *HTML* que estrutura a página das informações descritas. Consequentemente gera maior organização e torna possível realizar a formatação dos elementos que formam uma ou várias páginas de uma vez.

Podemos fazer uma analogia de como os dois funcionam. Imagine a construção de uma casa, pode-se assimilar que as paredes dessa casa torna-se a estrutura do *HTML*, e consequentemente para o ambiente tornar-se vistoso aplicaria todo o revestimento e pintura, dessa maneira conseguimos denominar como o *CSS*.

2.2.1. Regra @Font-Face

A linguagem de marcação *CSS* fornece diversos recursos que permite a manipulação dos textos que estarão contidos em uma página *HTML*, um desses recursos chamado *@font-face* é responsável pela manipulação das fontes que podem ser incorporadas através da família de fonte no *CSS*, essa fonte incorporada oferece um estilo de fonte personalizada.

Segundo [Mazza 2014] essa funcionalidade permite uma ampla flexibilidade para os desenvolvedores de páginas, que conseguem através desse recurso definir novas famílias de fontes.

O recurso para incorporar a fonte na folha de estilo é essencial no funcionamento dessa biblioteca por estar realizando o papel de manter os ícones dentro de sua estrutura e incorporando diretamente na folha de estilos quando chamado.

2.3. SVG - Scalable Vector Graphics

Os desenhos elaborados para representar cada situação, ou ação foram todos criados através do software de desenho vetorial no formato SVG que proporciona traços formosos e garante a qualidade na renderização.

Segundo a [W3C b] o SVG é uma linguagem que descreve gráficos vetoriais escalonável e vetores baseados em XML que podem ser incorporado em uma página HTML.

Os arquivos SVG possui uma ótima funcionalidade de não perder a qualidade da imagem quando necessário redimensionar. Portanto quando houver a necessidade de ampliar uma figura na tela através da biblioteca não perdera qualidade em seus traços.

2.4. Desenvolvimento

O desenvolvimento da biblioteca exigiu bastante dedicação e concentração na criação de cada ícone, pois, em muitos casos, foi necessário transformar desenhos grandes em miniaturas sem perder as principais características. Os ícones, mesmo bem pequenos, precisam ter seus formatos bem desenhados e definidos para representarem de maneira clara e objetiva o que propõem.

A estrutura do projeto é definida por dois arquivos, a folha de estilo CSS e a fonte. A fonte armazena cada ícone no formato SVG em uma posição interna de sua estrutura e disponibiliza um código de referência.

O segundo arquivo importante é a folha de estilo que disponibiliza e define todas as ações da biblioteca, nela está contida todas as classes que podem ser chamadas na estrutura do HTML, como os nomes dos ícones, os tamanhos, as cores, os efeitos e configurações próprias da biblioteca. Então quando o arquivo CSS é incorporado por uma página suas propriedades e regras são lidas e executadas, uma dessas regras é a *@font-face* que fica responsável por ler a fonte e conseguir acesso aos ícones.

Com o acesso concedido aos ícones às classes foram definidas com um nome que representam-as, quando a classe da biblioteca for invocada no código de uma página automaticamente o navegador irá ler os arquivos e renderizar o ícone na tela.

3. Funcionalidades

A aplicação dos ícones pode ocorrer em diversos componentes em uma página Web, possibilitando a liberdade para o desenvolvedor aplicar e ver o melhor posicionamento. As personalizações podem ocorrer em listas, botões, *links*, títulos ou outros componentes. Além disso, a biblioteca permite a utilização dos ícones juntamente com outros *frameworks* como o *Bootstrap* ou uma personalização própria de seu arquivo *CSS*. Além dos ícones oferecidos pela ferramenta, é possível utilizar outros recursos, como: cores, tamanhos e efeitos.

A acessibilidade na Web é de extrema importância, pois permite que pessoas com limitações, por meio de softwares, naveguem e acesse páginas Web. Por conta disso, é de extrema importância utilizar o atributo `aria-hidden="true"`, como

informado pela [W3C a], sobre a *tag* `<i>` para que os softwares de leitura de tela ignorem o ícone no momento da leitura do texto presente na página.

3.1. Instalação da Biblioteca

A biblioteca de ícones permite com que o desenvolvedor utilize de maneira simples e prática. Para começar, basta baixar o *pack* da ferramenta e realizar a implementação no projeto. Comece pela importação do *CSS* por meio da *tag* `<link>` dentro das *tags* `<head>...</head>`. Por exemplo:

```
<head>
...
<link rel="stylesheet" type="text/css" href="fp-icons.css">
...
</head>
```

Após a importação da biblioteca os ícones já estão prontos e disponíveis. É necessário criar uma *tag* `<i>` com as classes `fp` e `fp-nome`, em que `nome` representa o nome do ícone. O código abaixo é necessário para adicionar o ícone "blabla":

```
<i class="fp fp-blabla"></i>
```

O ícone pode ser colocado dentro de listas, ícones, títulos, parágrafos ou qualquer outra estrutura existente no HTML.

3.2. Variedade de Ícones

Com uma ampla variedade de ícones projetados, esta biblioteca disponibiliza os ícones divididos em seções que podem ser identificadas por: ícones da Web, pagamentos, áreas de atuação, acessibilidade, veículos, clima, tecnologias, objetos e outras representações.

A Figura 1 ilustra ícones relacionados com meios de pagamentos. Destaca-se que alguns desses ícones são de bancos ou meios de pagamentos que são nativos do Brasil e não são disponibilizados em outros *frameworks*.



Figura 1. Ícones de pagamentos disponíveis.

A Figura 2 apresenta ícones relacionados ao clima, como: sol, chuva, nublado e outras situações que podem ocorrer. Já a Figura 3 ilustra alguns *smiles*, comumente presentes no dia a dia em aplicativos de conversas e redes sociais.



Figura 2. Ícones de climáticos disponíveis.



Figura 3. Ícones de sorrisos e expressões disponíveis.

3.3. Cores

As cores disponibilizadas como padrão são baseadas no *Bootstrap* e oferecem para o desenvolvedor uma ampla gama de cores simpáticas e atraentes possibilitando ao desenvolvedor personalizar os ícones com sua aplicação. Elas podem ser alteradas ao inserir a classe `fp-color-nome` com a cor desejada na *tag* `<i>`. A Figura 4 ilustra as cores presentes nesta biblioteca e seus respectivos nomes. O código a seguir demonstra como seria uma implementação de um ícone com cor azul:

```
<i class="fp fp-heart fp-color-primary"></i>
```

- ♥ fp-color-primary
- ♥ fp-color-secondary
- ♥ fp-color-success
- ♥ fp-color-danger
- ♥ fp-color-warning
- ♥ fp-color-info
- ♥ fp-color-light
- ♥ fp-color-dark

Figura 4. Lista de cores e nomes

3.4. Tamanhos

Para alterar o tamanho de um ícone basta realizar a chamada da classe **fp-size-tamanho**. Assim, o ícone utilizado poderá ficar no tamanho desejado pelo desenvolvedor. Os tamanhos padrões disponíveis são ilustrados pela Figura 5 e variam de 2 a 8. Além dessa variedade de tamanhos é possível realizar a personalização própria pelo desenvolvedor com o CSS. O código abaixo demonstra como colocar um ícone com o tamanho 2:

```
<i class="fp fp-github fp-size-2"></i>
```

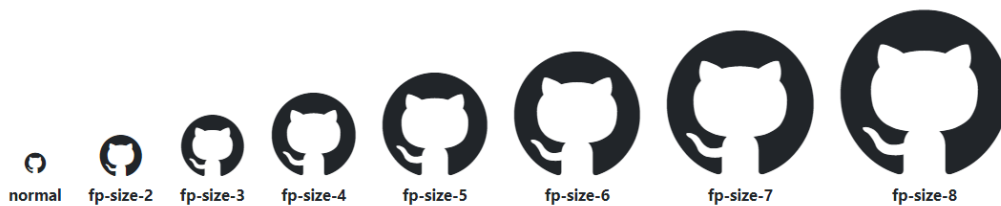


Figura 5. Lista de tamanhos

3.5. Efeitos e Animações

As animações possibilitam com que os ícones interajam com algumas animações na tela onde serão aplicados, como girar ou pulsar. A rotação de um ícone e a classe que representa essa rotação é ilustrada pela Figura 6. No total há seis rotações disponíveis, mas, assim como as cores, o desenvolvedor também tem a possibilidade de utilizar alguma outra que desejar. A classe **rotate-angulo** aplica os ângulos disponíveis e o código a seguir demonstra como girar o ícone a 90 graus.

```
<i class="fp fp-smile-happy fp-rotate-90"></i>
```

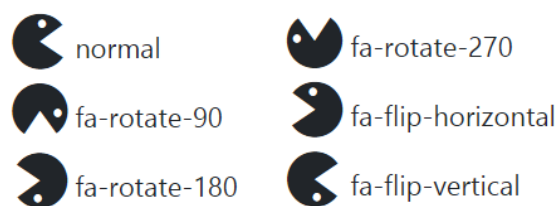


Figura 6. Lista de ângulos de rotação

O efeito de piscar pode ser utilizado de duas formas: se deseja que o ícone pisque apenas quando carregar a página, chame apenas a classe **blink**; caso deseje que ele fique piscando infinitamente chame duas classes **blink** e **infinite**. É ilustrado pela Figura 7 como seria seu funcionamento em *frames*, ou seja, qual variação o ícone iria receber até voltar em sua forma inicial. O código abaixo demonstra como seria a utilização em código para deixar um ícone piscando sem parar.

```
<i class="fp fp-github blink infinite"></i>
```




Figura 7. Efeito blink em frames de seu funcionamento

3.6. Utilização em Listas e Botões

A personalização em listas pode ocorrer de maneira bem simples. Para isso, deve-se adicionar a classe `fp-ul` na `tag <ul>` e a classe `fp-li` em cada `<li>` dessa lista. A Figura 8 ilustra o resultado do código abaixo:

```
<ul class="fp-ul">
  <li><i class="fp-li fp-circle"></i>Seu texto</li>
  <li><i class="fp-li fp-checked"></i>Seu texto</li>
  <li><i class="fp-li fp-heart"></i>Seu texto</li>
  <li><i class="fp-li fp-youtube"></i>Seu texto</li>
</ul>
```

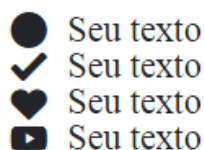


Figura 8. Exibição da implementação da lista

A utilização em um botão também é fácil: adicione as classes `fp` e `fp-nome-do-icone` na classe no botão. A Figura 9 demonstra como ficaria um botão com o ícone do *YouTube* e um texto e outro botão somente com um ícone de coração, que foram criados a partir do código abaixo:

```
<button class="fp fp-youtube"> Meu botão</button>
```

```
<button class="fp fp-heart"></button>
```

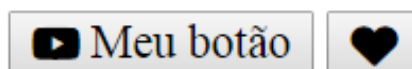


Figura 9. Exibição do botão com ícone

3.7. Personalização

Existe a possibilidade de personalização com outras cores e tamanhos conforme a necessidade por parte do desenvolvedor. Para colocar a cor personalizada basta criar uma nova classe em seu arquivo CSS contendo uma definição de cor e utilizar a classe personalizada na `tag` do ícone desejado. O código abaixo exemplifica como ficaria uma classe de cor personalizada em um ícone:


```
.minhaClasseCor{
    color:green;
}

<i class="fp minhaClasseCor fp-heart"></i>
```

A personalização do tamanho também segue a mesma ideia. Para isso, cria-se no CSS uma classe contendo um tamanho e aplica-se na classe do ícone. O código a seguir exemplifica uma situação dessa:

```
.minhaClasseTamanho{
    font-size:12em
}

<i class="fp minhaClasseTamanho fp-heart"></i>
```

Também é possível realizar a utilização de um ícone através do código *unicode*. Para isso, basta seguir alguns passos: criar no CSS a classe contendo o pseudo elemento que exibirá o ícone antes (*::before*) ou depois (*::after*); utilizar a propriedade *font-family* com o nome da fonte do *pack* 'FP'; inserir a propriedade *content* com o código *unicode* do ícone.

A seguir é exibido um código de exemplo em que antes de todas as *tags* `<span></span>` será exibido o ícone com o código *unicode* `\E013`. Nesse caso, o coração irá aparecer conforme o conteúdo exibido pela Figura 10.

```
span::before {
    font-family: 'FP';
    content: "\E013";
}
```

```
<span>texto</span>
```

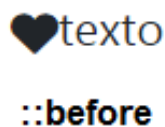


Figura 10. Implementação personalizada através do unicode

4. Considerações Finais

É de grande valia os resultados obtidos no desenvolvimento da biblioteca, na simplicidade quando implementada em uma página Web. Tem com uma de suas principais vantagens disponibilizar uma variedade de ícones de maneira gratuita, ícones que representam situações ou soluções nativas do Brasil, como os meios de pagamentos. Além disso possibilita a utilização juntamente com outras bibliotecas ou *frameworks* abrangendo um número maior de possibilidades durante o desenvolvimento de aplicações.

Uma de suas limitações é de estar disponível apenas na folha de estilo CSS tornando assim o uso apenas em páginas Web, não podendo ser aplicadas em tecnologias que não utilize estruturação HTML.

As melhorias futuras poderão ser realizadas para que a biblioteca funcione e atenda as linguagens de estilos dinâmicas *Sass* e *Less*. E ser dividida por seções para que os desenvolvedores baixem apenas as seções que mais atenda a necessidade em seu projeto. O projeto final está disponível para outros desenvolvedores utilizarem no *github*.

Referências

- Marques, C. D. (2016). A arte rupestre | monções revista do curso de história da ufms/cpcx. Disponível em: <https://periodicos.ufms.br/index.php/moncx/article/view/3126/2530>. (Acessado em 22/07/2019).
- Mazza, L. (2014). *HTML5 e CSS3: domine a web do futuro*. Editora Casa do Código.
- Miyagusku, R. H. M. (2007). *Desvendando os recursos do CSS*. Universo dos Livros Editora, 1º edition.
- Nielsen, J. (1993). *Usability engineering*. Morgan Kaufmann, 1º edition.
- W3C. Accessible rich internet applications (wai-aria) 1.1. <https://www.w3.org/TR/wai-aria-1.1/#aria-hidden>. (Accessed on 28/07/2019).
- W3C. Scalable vector graphics (svg) 2. <https://www.w3.org/TR/SVG2/>. (Acessado em 23/07/2019).