

Terceira dinâmica

Leonardo Guarnieri de Bastiani

Este documento trata a componetização com Web Components de duas telas apresentados no processo seletivo para a Casa e café 2017.

Tela 1: Assinar Premium

Assinando premium você ganhará maior relevância para que os melhores profissionais te contatem primeiro poderá, além de poder conversar com quantos profissionais desejar e ver recomendações completas dos candidatos.

Cupom de Desconto: Adicione seu Cupom Aqui

Total: R\$84,00

Método de Pagamento: ☐ Cartão ☐ Boleto

Se ainda estiver com dúvidas ou precisar de ajuda, é só falar com a gente. [Fale Conosco](#)

Tela 2: Pagamento no Boleto

TOTAL A PAGAR: R\$ 89,90

CPF do Titular:

Pagamentos com Boleto Bancário demoram até três dias para serem validados. Use seu cartão de crédito e ative o Premium agora mesmo!

Vencimento: dd/mm/aaaa

[Gerar Boleto](#)

Ao realizar o pagamento, você concordará com [nossos termos e condições de uso](#).

Tela 1.

Tela 2.

Os componentes foram escolhidos seguindo os critérios:

- Os componentes se repetem e a utilização de *Web Components* facilita a reutilização de código.
- A modularização, importante para o entendimento do projeto, aumento da divisão de tarefa entre a equipe e testes separados.

A seguir, estão apresentados os componentes:

Grid

A disposição dos elementos na tela em tabela pode ser componetizada com um elemento <grid> que seria semelhante a uma <table> em HTML. O grid é organizado em linhas e colunas. A Fig. 1 exemplifica a disposição do elemento grid na página. A ideia foi inspirada nos Tiles do Bulma (<https://bulma.io/documentation/layout/tiles/#example>).

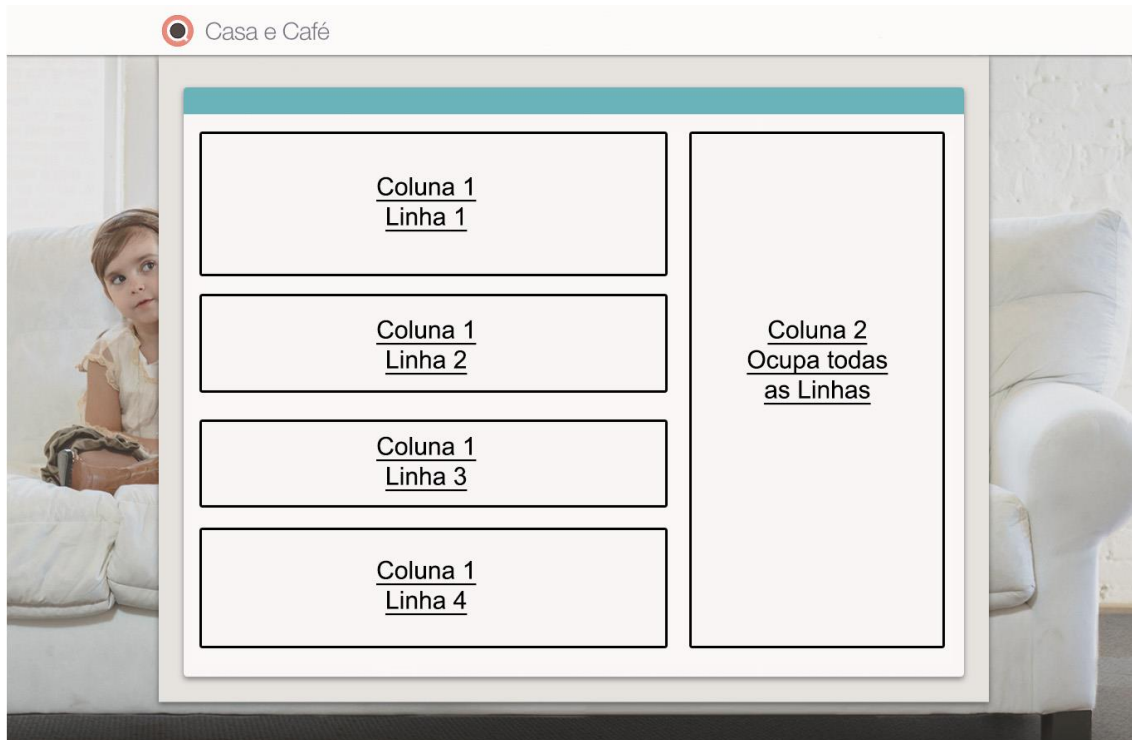


Figura 1. Exemplificação da exibição do Web Component Grid.

Além disso, o grid deve inserir a divisória entre os subelementos dele. A Fig 2 é idêntica a Fig. 1, porém ela contém essas divisórias que estão entre as colunas e linhas, o detalhe é pequeno.

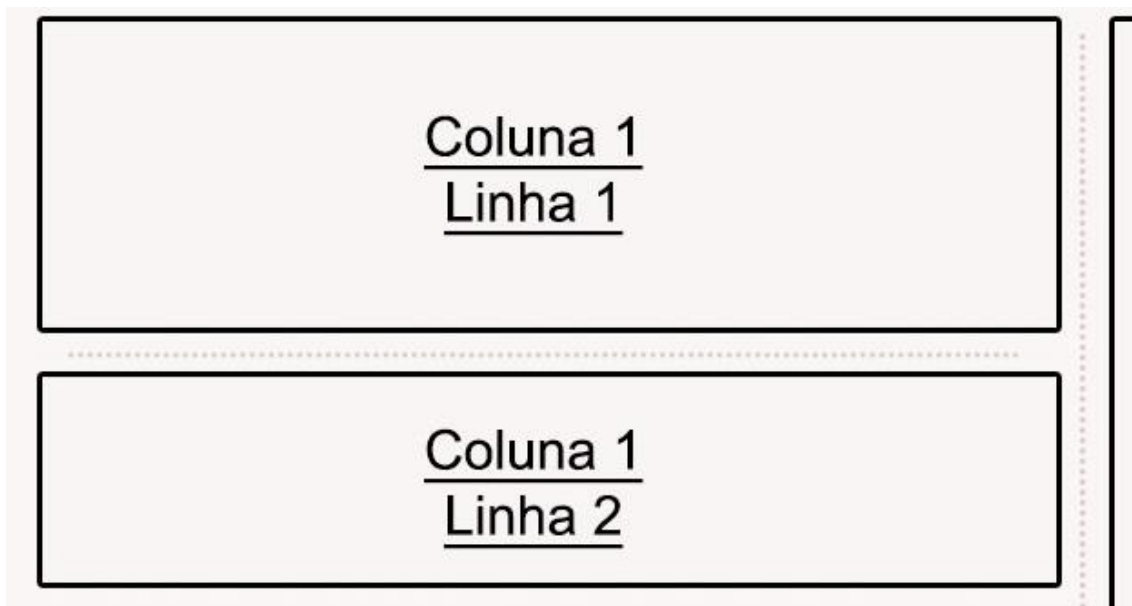


Figura 2. Web Component Grid com as divisões inseridas. É um recorte da Fig. 1.

Possui as funcionalidades de aumentar a compatibilidade entre navegadores, pois interpretação do código Web Component é feita em Javascript, e simplificar o código em HTML.

Fade

Este Web Component foi inspirado na biblioteca jQuery de transição de elementos. (https://www.w3schools.com/jquery/jquery_fade.asp).

Nas telas de referência, é imaginável que o pagamento em cartão e o pagamento em boleto transitem, como por exemplo fade in e fade out, este Web Component que poderia ser implementado com a tag <fades> e dentro a tag <fade> e trocava estes elementos seguindo a sequência de quadros da Fig. 3.

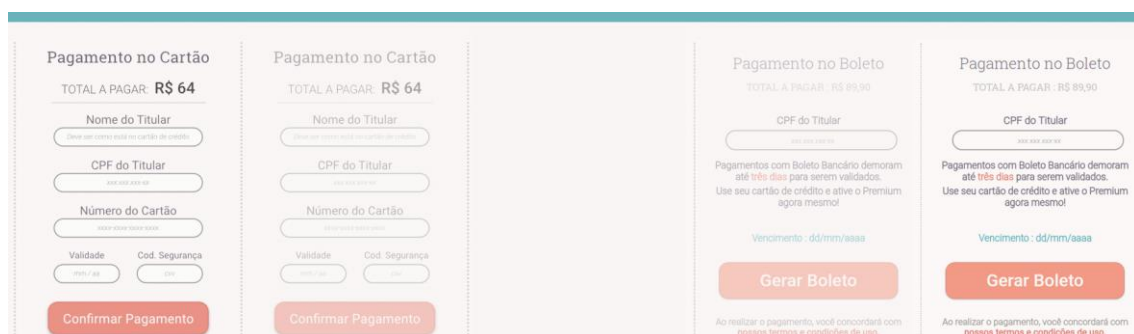


Figura 3. Transição entre telas com o Web Component Fade.

A criação de um Web Component para este caso serviria tanto para modularização como para reutilização.

Input-mask

Este componente foi inspirado nos elementos de inserção de dados com máscara, ou seja, que possuem uma regra de restrição para os dados que podem ser inseridos; e por este site <https://css-tricks.com/input-masking/>.

O elemento poderia ser utilizado com a tag `<input-mask>` e é uma extensão da tag `<input>`. A Fig. 4 apresenta o Web Component de como ele foi idealizado junto com uma `<label>`.



Figura 4. Exemplo de um `<input-mask>`.

Além da restrição, o `<input-mask>` aumenta a UX, pois o usuário percebe o que deve ser inserido, o que falta e o que não pode ser inserido.

Radio-button

Este Web Component possui a funcionalidade de uma tag `<input>` do tipo "radio", mas com a exibição de botões em vez do padrão apresentado pelos navegadores. Foi inspirado nos Radio buttons do Bootstrap (<https://getbootstrap.com/docs/4.0/components/buttons/#checkbox-and-radio-buttons>).

A mudança de estado desses elementos pode ser tratada por uma Callback em Javascript.

A Fig. 5 apresenta a exibição do Web Component na página. Ele poderia ser inserido no HTML com a tag `<radio-button>`.



Figura 5. Web Component Radio-button.

Money

Este Web Component foi idealizado para exibir valores em dinheiro. Não possui nenhuma inspiração encontrada previamente na Web, mas pode ser considerado um componente importante devido ao erro de ponto flutuante em Javascript, além disso, ele pode ser utilizado para a conversão de moeda.

Ele foi planejado para ser utilizado com a tag `<money>` e, dentro dela, as tags `<prefix>`, `<value>`, `<dot>` e `<decimal>`. O Código 1 apresenta um exemplo

da utilização dessa tag, e se houvesse uma implementação correta do CSS, este código resultaria na figura 6. A figura 7 também é um exemplo de utilização do Web Component Money, porém sem as tags <dot> e <decimal>.

Código 1. Exemplo da utilização do Web Component Money.

```
1 <money>
2   <prefix>R$</prefix>
3   <value>84</value>
4   <dot>,</dot>
5   <decimal>00</decimal>
6 </money>
```

A imagem mostra o resultado da aplicação do CSS ao código HTML. O texto "R\$84,00" é exibido em uma fonte serifada, com o "R\$" em um tamanho menor e a cor cinza, e "84,00" em um tamanho maior e cor verde-azulado.

Figura 6. Resultado do Código 1 após a aplicação do CSS que não foi apresentado.

A imagem mostra o resultado da aplicação do CSS ao código HTML. O texto "TOTAL A PAGAR: R\$ 64" é exibido em uma fonte sans-serif, com "TOTAL A PAGAR:" em um tamanho menor e cor cinza, e "R\$ 64" em um tamanho maior e cor verde-azulado.

Figura 7. Exemplo do Web Component Money exibindo um valor sem a precisão decimal.

Este Web Component pode ter parâmetros para facilitar o programador Javascript como alteram a exibição, como a aplicação de desconto, tanto em porcentagem como em valores.

Conclusão

Os Web Components apresentados foram inspirados em Frameworks conhecidos como o Bootstrap, Bulma e o Materialize; ou foram inspirados para solucionar problemas comuns no ramo do Web Design.

Vale ressaltar que as telas exibidas no projeto possuem elementos do tipo <input> que são muito semelhantes e que se repetem com frequência. Essa característica não foi abordada aqui por um Web Component, pois isso poderia ser solucionado facilmente com a adição de uma Folha de Estilo (CSS).