



TERCEIRA DINÂMICA

Processo Seletivo Casa e Café 2018

Caroline Castor dos Santos
carolinecdsantos@gmail.com

Objetivo

Como terceira etapa da dinâmica do processo seletivo da Casa e Café, nos foi proposto que analisássemos os seguintes PrintScreens de uma página do site da empresa e definíssemos quais campos poderiam ser candidatos a WebComponent.

Casa e Café

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

Pagamento no Cartão

TOTAL A PAGAR: R\$ 64

Nome do Titular
Deve ser como está no cartão de crédito

CPF do Titular
XXX.XXX.XXX-XX

Número do Cartão
XXXX-XXXX-XXXX-XXXX

Validade
MM/YY / MM/YY

Cod. Segurança
CVV

Confirmar Pagamento

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

Figura 1 - Tela de Assinatura Premium

Pagamento no Boleto

TOTAL A PAGAR : R\$ 89,90

CPF do Titular
XXX.XXX.XXX-XX

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

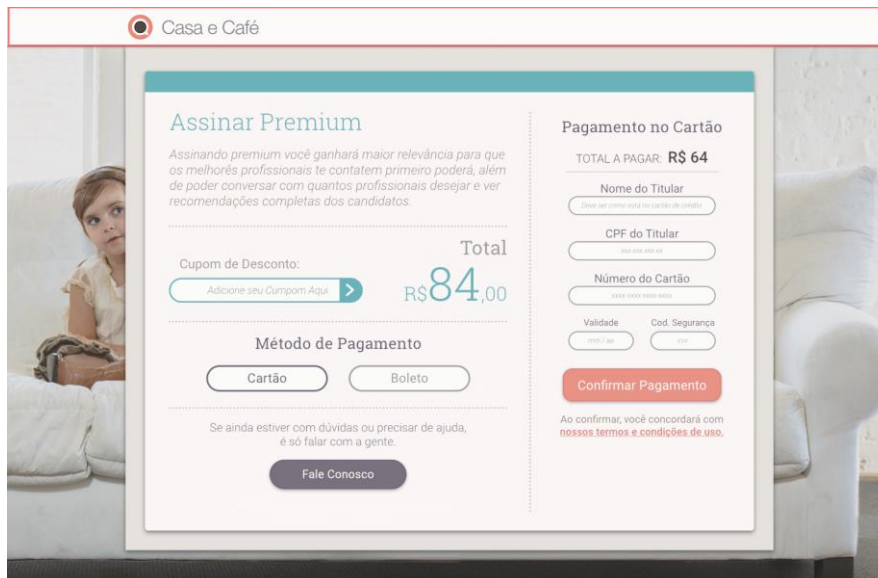
Ao realizar o pagamento, você concordará com [nossos termos e condições de uso.](#)

Figura 2- Desdobramento da Figura 1 opção boleto

Análise

WebComponents devem ser aplicados de forma a reutilizar código, facilitar a localização e alteração de um componente web, tornando o código mais legível e compacto. Sendo assim, identifiquei os seguintes elementos a serem candidatos:

Header da Página



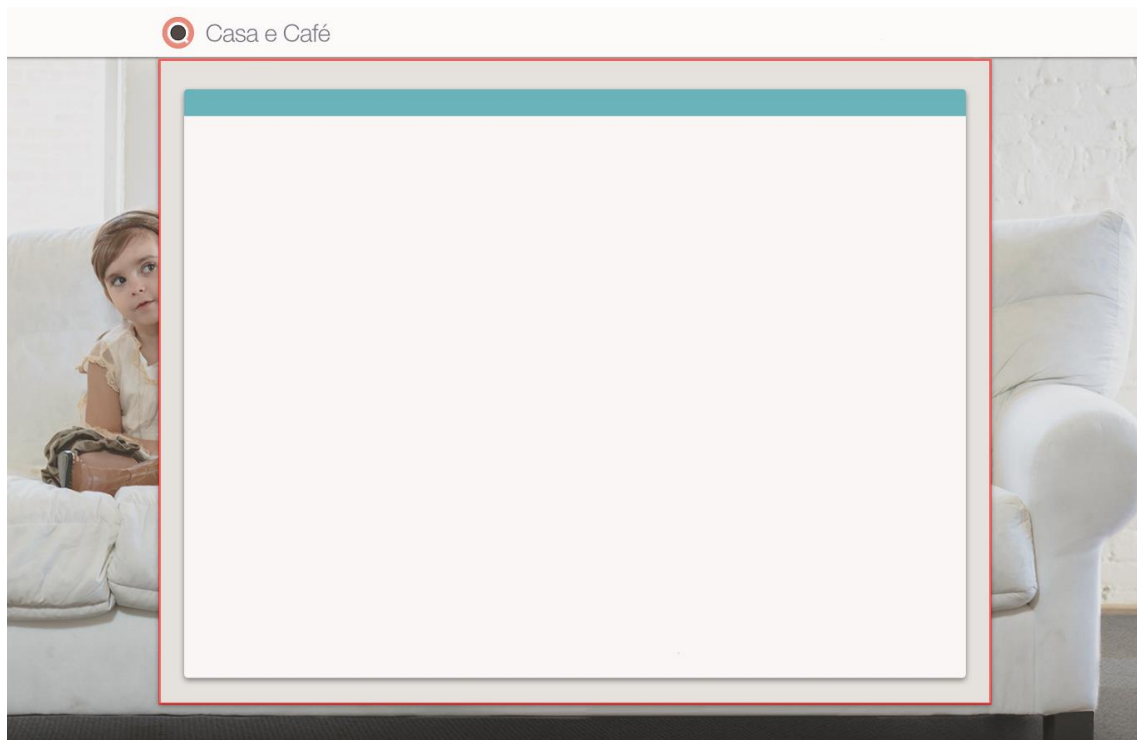
Por que deveria ser um WebComponent?

O header geralmente está presente em todas as páginas, então atuaria como um componente de reuso. Poderíamos pensar em uma tag do tipo:

```
<header>
</header>
```

Algumas funcionalidades interessantes seriam: a escolha da imagem do logo, cor de fundo e também a possibilidade de utilização de outras tags para a criação de um menu contido no header por exemplo. Além disso, o estilo aplicado não entraria em conflito com outros estilos que poderiam ter a mesma classe e ficaria tudo encapsulado.

Div de Destaque de conteúdo



Por que deveria ser um WebComponent?

É uma div de destaque que poderia reutilizar no código com outros contents dentro dela.

Provavelmente esse div é composto de outros divs mais internos como por exemplo a borda, o conteúdo que aparecerá dentro e dividido em partes e etc, ou seja, um aglomerado de divs que provavelmente será utilizado mais de uma vez para outras telas da página web.

Exemplo: Poderia utilizar esse mesmo estilo de div para tela de login, cadastro e etc. deixando a página mais padronizada e agradável.

Nessa componentização poderíamos especificar a cor da borda de acordo com as cores usadas na página, transformando em um objeto personalizável.

Cupom de Desconto



Por que deveria ser um WebComponent?

Ao observarmos esse campo podemos notar que o estilo é diferenciado, é constituído de label e input, além disso, o posicionamento dos elementos que o compõe torna-o único.

O elemento em si já carrega uma funcionalidade que é gerar um evento de validação de cupons para a aplicação imediata do desconto no preço de compra. Estendendo nosso pensamento, poderíamos utilizá-lo como WebComponent de qualquer validação que precisasse ser imediata na

página.

Exemplo: Um bom reuso desse tipo de input seria a informação de um CEP e preenchimento automático do endereço do usuário. O princípio é o mesmo, porém a validação é diferente. Como a validação é feita geralmente por uma requisição à API, esse componente teria os parâmetros da requisição e o endereço da API por exemplo. Em resumo, esse componente encapsularia essa requisição e estilo do componente para evitar conflito de estilo com outras classes.

Formato Sugerido:

```
<input-validate>
</input-validate>
```

Formulário de Pagamento com cartão

Pagamento no Cartão

TOTAL A PAGAR: R\$ 64

Nome do Titular
Deve ser como está no cartão de crédito

CPF do Titular
000.000.000-00

Número do Cartão
0000-0000-0000-0000

Validade
mm / aa

Cod. Segurança
CVV

Confirmar Pagamento

Ao confirmar, você concorda com [nossos termos e condições de uso.](#)

Por que deveria ser um WebComponent?

A criação de formulários requer muitas tags misturadas com requisições, validações e etc. Nesse caso, temos um formulário de pagamentos com cartão de crédito que basicamente em todas as aplicações tem os mesmos campos como Nome do Titular, CPF do titular, Número do Cartão de Crédito, CVV e validade, além de uma requisição à API para validar tudo isso. Tornando essa formulário, toda essa parte de estilo, definição dos campos elementares a um formulário de pagamento, requisição à API estaria encapsulada deixando o código mais enxuto e permitindo que o usuário colocasse algumas entradas adicionais se for o caso.

Formato Sugerido:

```
<form-payment>
  (... quando adicionado já carrega os campos defaults do pagamento)
  (...tags de input adicionais)
</form-payment>
```

Estilo de Botões Padronizados



Por que deveria ser um WebComponent?

Podemos observar que o mesmo estilo de botão é utilizado várias vezes na página (de forma a manter o padrão). Sendo assim, é justificável que esse estilo de botão seja um WebComponent para evitar repetição de código e conflitos na estilização de outros components.

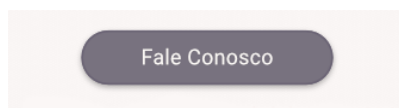
Estilo de Input Text Padronizado

Nome do Titular	CPF do Titular	Número do Cartão	Validade	Cod. Segurança
<small>Deve ser como está no cartão de crédito</small>	<small>xxx.xxx.xxx-xx</small>	<small>xxxx-xxxx-xxxx-xxxx</small>	<small>mm / aa</small>	<small>cvv</small>

Por que deveria ser um WebComponent?

Podemos observar que o mesmo estilo de input text é utilizado várias vezes na página de forma a manter o padrão. O que é mutável é somente a validação e a máscara. Sendo assim, é justificável que esse estilo de input seja um WebComponent para evitar repetição de código e conflitos de outros componentes.

Botão de Fale Conosco



Por que deveria ser um WebComponent?

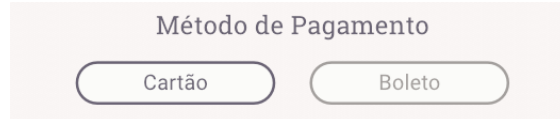
Ao observar esse elemento, imagino que ao clicarmos nesse button, seria aberto um modal com algum formulário de contato com a empresa ou uma outra página. Formulário de contato com a empresa geralmente tem os mesmos campos (Nome, E-mail, Mensagem) e talvez alguns campos opcionais que poderiam ser inclusos pelo desenvolvedor. Sendo assim poderíamos pensar em uma tag do tipo:

```
<form-contact>  
  (campos default já inclusos na declaração do elemento desse tipo)  
  (campos adicionais)  
</form-contact>
```

Colocando em um modal ou até mesmo em uma página separada de acordo com a preferência do desenvolvedor.

Considerações Finais

Grupo de Botões de Seleção de Pagamento



Método de Pagamento

Cartão Boleto

Porque NÃO deveria ser um WebComponent?

Ainda na imagem 1, o grupo de botões que faz a seleção entre as opções de pagamento por cartão ou boleto não seria um candidato à WebComponent por ser um elemento dinâmico que ao ser selecionado uma das duas opções, altera a segunda parte da dela de acordo com a opção selecionada. Além disso, a construção é simples e pouco reutilizável em outras partes da aplicação.

Formulário de Pagamento por Boleto Bancário



Pagamento no Boleto

TOTAL A PAGAR : R\$ 89,90

CPF do Titular

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.

Porque NÃO deveria ser um WebComponent?

Na segunda tela, que é um desdobramento da primeira ao se selecionar a opção boleto, o formulário não deve ser um WebComponent, pois pensando em internacionalização da página, a opção boleto só seria aplicável no Brasil, além de só conter uma informação requerida que é um outro WebComponent sugerido. Sendo assim, na segunda tela, estaríamos reutilizando os WebComponents de input, botões.