



ALTERNATIVA DE LAYOUT PARA HTML COM BASE NO SISTEMA DE LAYOUT DINÂMICO DO WPF

Gustavo Rech

Orientador: Prof. João Batista S. de Oliveira

Ciência da Computação

gustavo.rech@acad.pucrs.br



HTML e CSS, as duas linguagens para descrever interfaces Web, são linguagens versáteis e poderosas, mas não é incomum nos depararmos com grandes dificuldades para atingirmos os resultados que almejamos. Como consequência disso o tempo de desenvolvimento pode ser aumentado e, muitas vezes, não conseguimos reproduzir da forma inicialmente projetada. Existe alguma forma de mitigar isso?

Motivação

Os navegadores foram projetados para disponibilizar documentos de forma digital e, nos primórdios, para disponibilizar artigos científicos entre universidades. O layout foi projetado para documentos, como é a ferramenta Word da Microsoft, mas logo a ideia de navegadores de internet expandiu, novos usuários começaram a utilizar, e as páginas passaram a disponibilizar documentos não mais restritos a artigos científicos. O layout precisava ser atualizado para comportar os novos conteúdos, e ele foi, mas em cima do que já existia.

Hoje, um termo muito pesquisado em ferramentas de busca na internet é "CSS tricks", truques para a utilização do CSS, e esse termo demonstra o que designers e desenvolvedores Web precisam, muitas vezes, para criar o layout de uma página, uma vez que para atingir um determinado layout nem sempre há algo declarativo e simples no CSS.

Objetivo

Este trabalho tem por objetivo disponibilizar, para pessoas que não se adaptam bem com o layout disponível no CSS, uma nova opção de declaração de layout, com a qual possam se sentir mais confortáveis e reduzir a quantidade de declarações necessárias para chegar no resultado esperado. Será definido e apresentado um **superset** de declarações sobre o HTML e CSS a fim de atingir uma solução de layout em que os elementos possuam um posicionamento baseado no auto **dimensionamento** e **alinhamento** de elementos, e que o **comportamento** dos elementos dependa majoritariamente do elemento pai.

As declarações de layout especificadas e o comportamento resultante serão semelhantes às de aplicações tradicionais pensadas em janela e painéis, e terão por base o sistema de layout dinâmico implementado pelo **WPF**. A solução de layout deverá funcionar em conjunto com toda a estrutura já existente do HTML e do CSS, com isso todo o poder de processamento gráfico já implementado e otimizado nos navegadores junto a declaração, interpretação e processamento do HTML e CSS serão aproveitados.

HtmlAlign

É um renderizador geométrico dinâmico feito em javascript que atua sobre o renderizador do HTML e, a partir de um conjunto de declarações feitas sobre o HTML e o CSS, realiza o cálculo de layout, posiciona os elementos na tela e detecta alterações de forma dinâmica.

Auto Dimensionamento

Todo elemento pode ampliar o seu tamanho para comportar a necessidade dos elementos internos.

Alinhamento Total

Todo elemento tem a liberdade de se posicionar como quiser no espaço entregue pelo elemento pai.

Comportamento no Elemento Pai

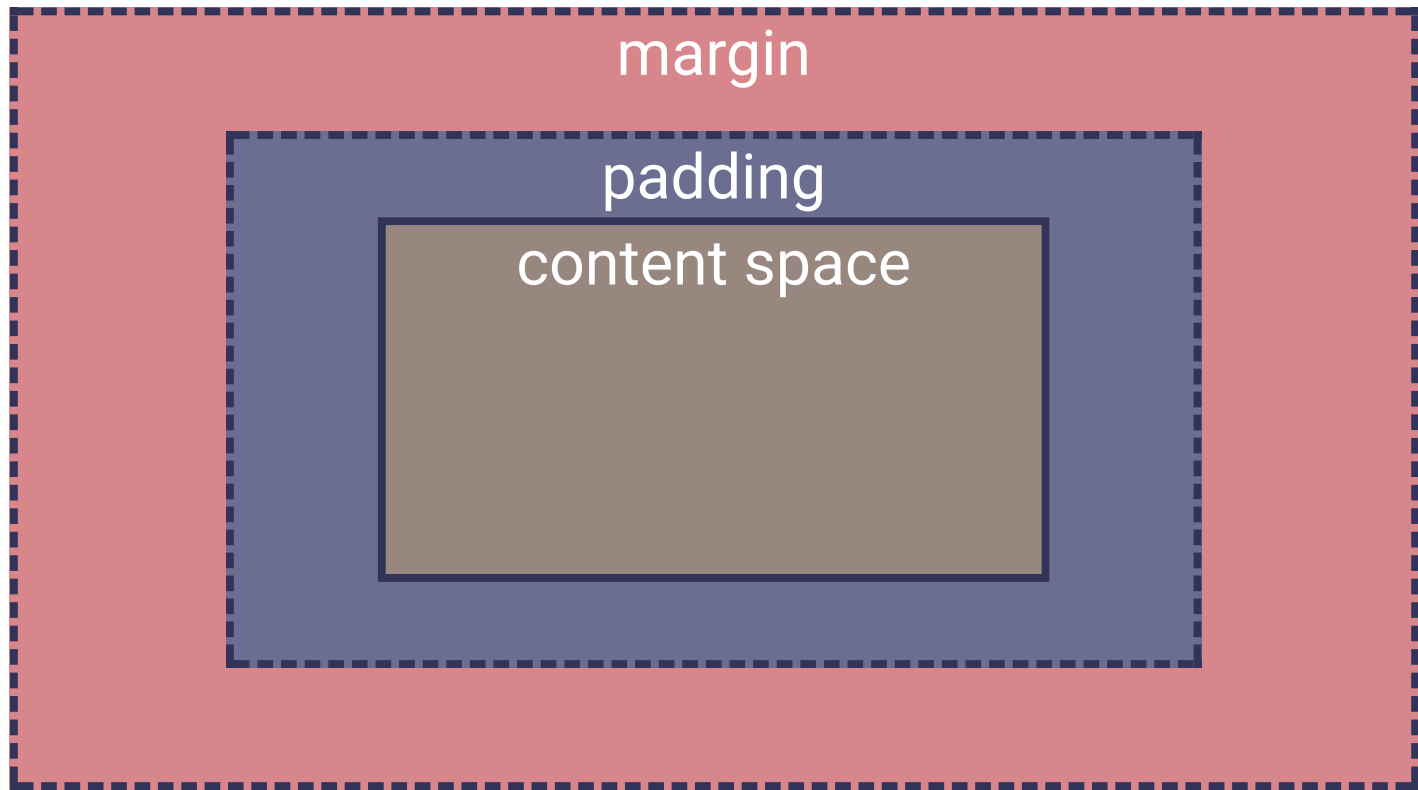
O elemento pai tem controle sobre os elementos filhos entregando espaços diferentes para cada um.

Extensível

Novos comportamentos e declarações podem ser adicionados.

Comportamento Base:

Modelo de Caixa:



Alinhamento:

	left	center	right	stretch
top				
center				
bottom				
stretch				

Delimitação de Tamanho:

(por dimensão)

Absoluto:

~ 200 ~400 200~ 200~400

Percentual:

50% ~50% 50%~ 200~50% 50%~400

Percentual Delimitado:

50%[~400] 50%[200~] 50%[200~400]

Proporcional:

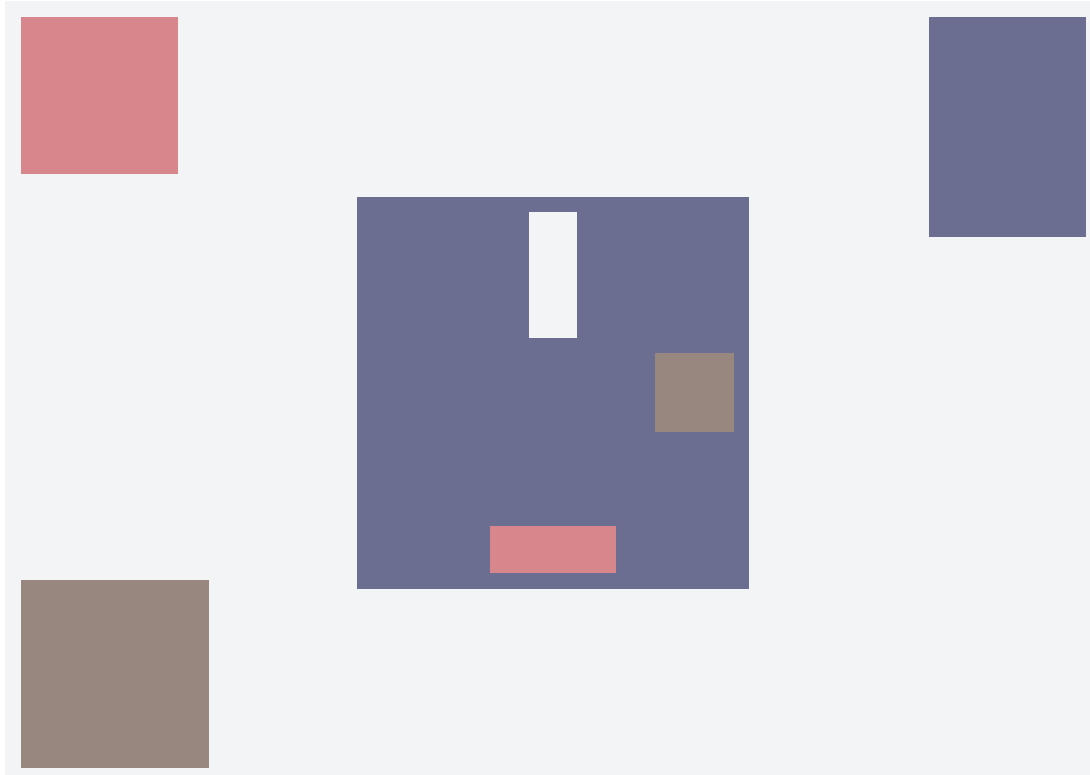
* 2* 4*

Declaração:

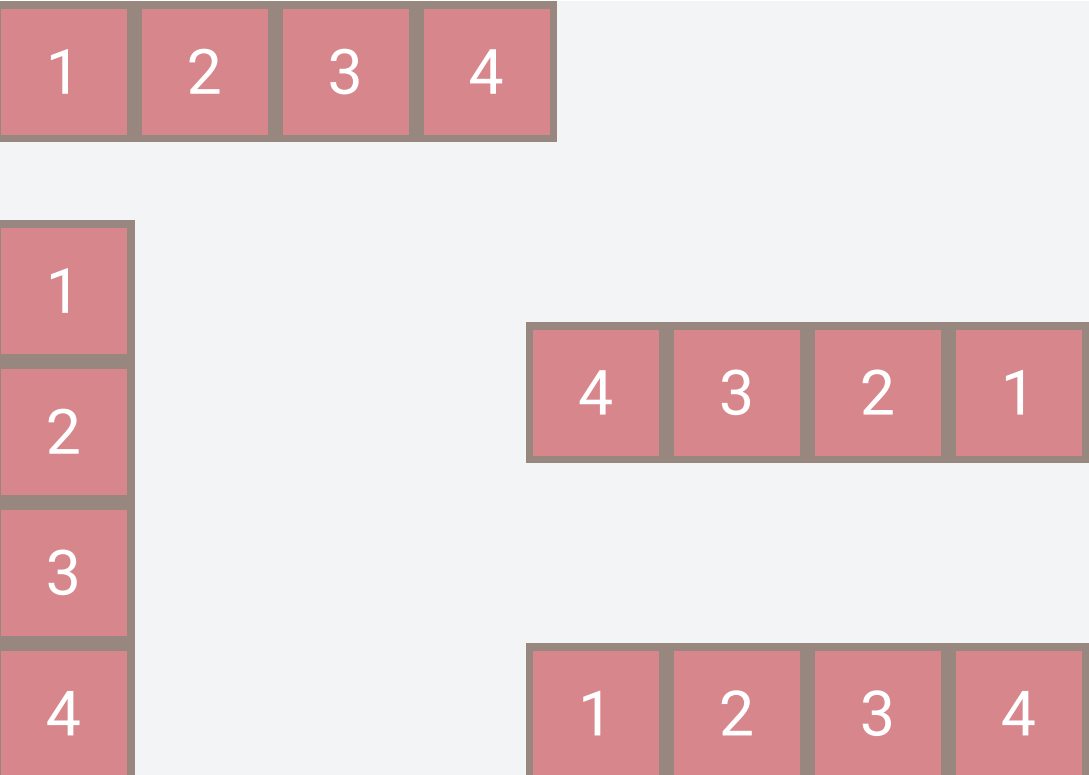
```
.wrapper {
  --behavior: stack;
  --stack: top;
  --align: top center;
  --size: 100%[600~1200] ~;
  padding: 20px;
}
```

Principais Comportamentos:

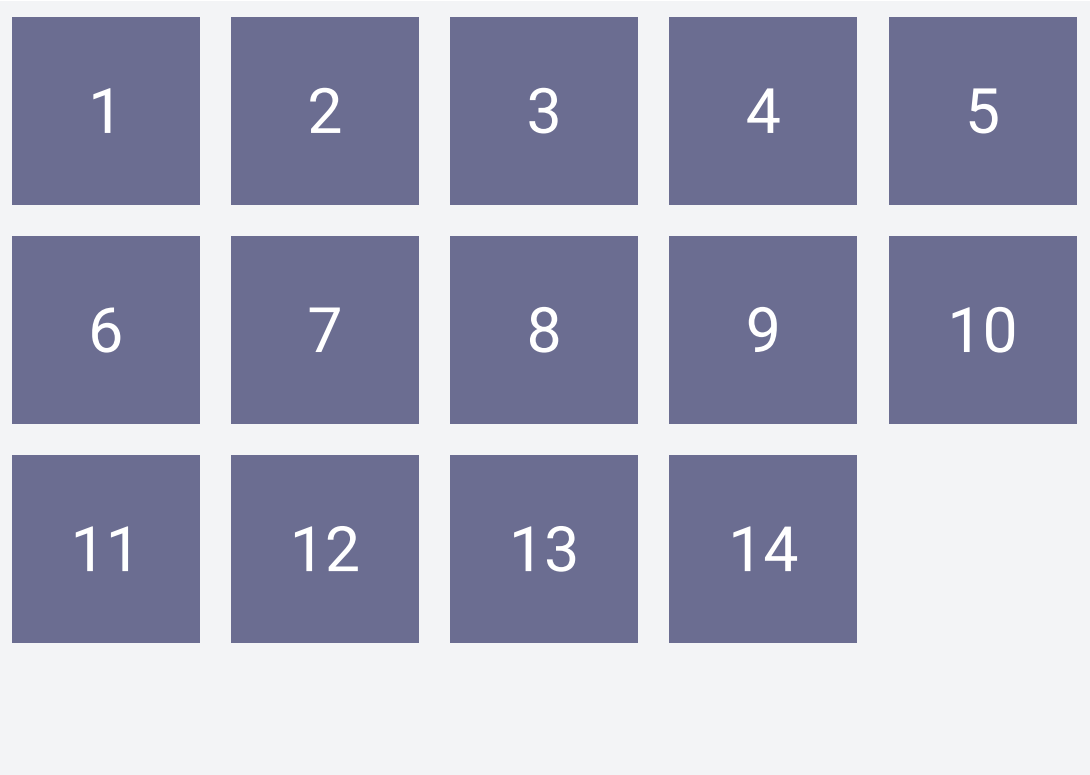
Panel:



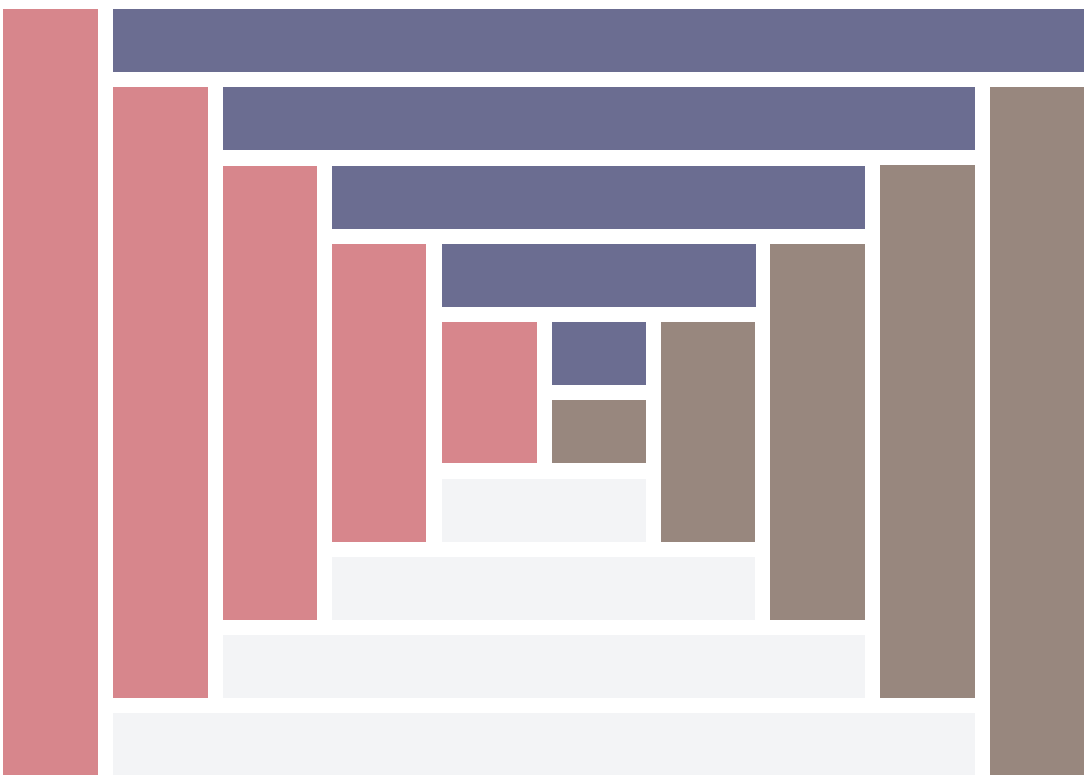
Stack:



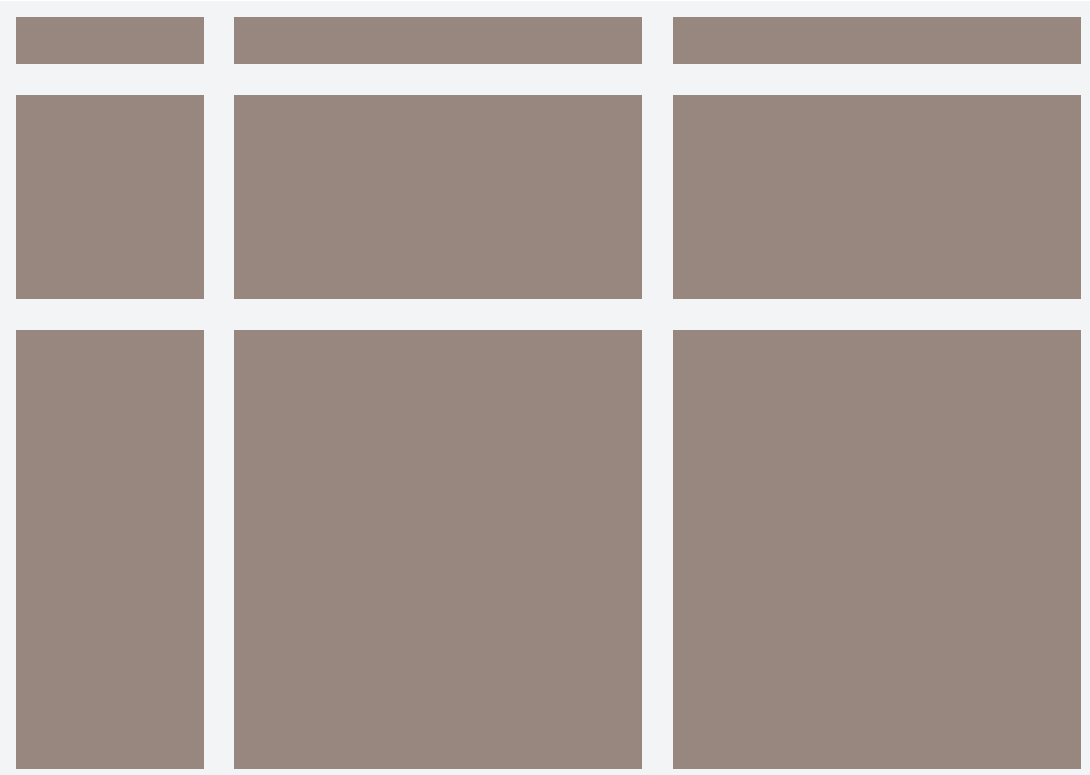
Wrap:



Dock:



Grid:



Este poster foi construído utilizando HTML, CSS e a implementação deste trabalho.

Para acessar a versão em HTML utilize o QR-Code ao lado. É necessário utilizar um browser compatível com **custom properties** e **max-content**. Os browsers **Chrome**, **Firefox**, **Opera** e **Safari** com versões mais novas já possuem essas features implementadas.

Infelizmente o **Internet Explorer** não é compatível em todas as suas versões. O **Microsoft Edge** ainda não é compatível, mas, segundo o site de acompanhamento de implementação do Edge, já está em processo de implementação.

<http://tiny.cc/64imly>

