



argonavis.com.br

SVG

3

estrutura do documento

grupos, símbolos e marcadores

reuso de objetos por referência

Helder da Rocha (helder@argonavis.com.br)



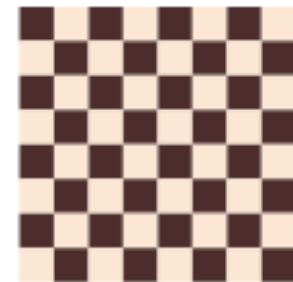
Estrutura de um documento

- Documentos SVG *podem* declarar figuras como elementos de primeiro nível logo abaixo do elemento `<svg>`
 - Há muita **repetição**: arquivos grandes e ineficientes

```
<svg xmlns:xlink="http://www.w3.org/1999/xlink"
     xmlns="http://www.w3.org/2000/svg" width="1050" height="600" viewBox="0 0 1050 600">
```

64 quadrados

```
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(20)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(40)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(60)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(80)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(100)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(120)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(140)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(160)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(20,20)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(40,20)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(60,20)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(80,20)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(100,20)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(120,20)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(140, 20)" />
...
<rect x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" transform="translate(140,140)" />
<rect x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" transform="translate(160,140)" />
</svg>
```



66 linhas
~3500 chars

- Ideal é **reusar** componentes: + eficiente, + fácil de manter/gerar
- SVG sugere uma estrutura para organizar o documento



Elementos estruturais

```
<svg ... viewBox="0 0 1400 900" <!-- Elemento <svg> -->
  width="1400px" height="900px" <!-- ViewBox inicial -->
  <script>...</script> <style>...</style> <!-- Scripts e CSS -->
  <defs> <!-- Bloco de definições -->
    <linearGradient id="arcoiris"> ... </linearGradient> <!-- Gradientes -->
    <filter id="desfoque"> ... </filter> <!-- Filtros -->
    <pattern id="grade"> ... </pattern> <!-- Padrões -->
    <marker id="seta"> ... </marker> <!-- Marcadores -->
    <g id=""> ... </g> <!-- Grupos -->
    <symbol id="diodo" > <!-- Símbolos -->
      <title>Diodo</title> <!-- Documentação -->
      <desc>Simbolo de diodo comum.</desc> <!-- Subcomponentes -->
      <rect x="4" y="0" ... />
      <polygon points="0,0 10,0 5,8" />
    </symbol>
    <symbol id="transistor"
      viewBox="-1 -1 22 22">... </symbol> <!-- viewBox local -->
  </defs>
  <use xlink:href="#diodo" <!-- Objetos referenciados -->
    x="50" y="50" height="25" width="25"/>
  <use xlink:href="#transistor"
    x="100" y="50" height="50" width="50"/> <!-- Grupos -->
  <g transform="translate(200,30) scale (-1,1)"> ... </g>
</svg>
```



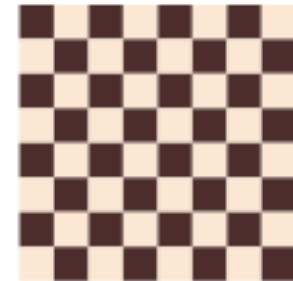
Documento estruturado

```
<svg xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.w3.org/2000/svg"
      width="1050" height="600" viewBox="0 0 1050 600">
  <defs>
    <rect id="preto" x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" />
    <rect id="branco" x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" />

    <g id="quatroCasas">
      <use xlink:href="#preto" />
      <use xlink:href="#branco" transform="translate(20)" />
      <use xlink:href="#branco" transform="translate(0,20)" />
      <use xlink:href="#preto" transform="translate(20,20)" />
    </g>

    <g id="fileiraDupla">
      <use xlink:href="#quatroCasas" />
      <use xlink:href="#quatroCasas" transform="translate(40)" />
      <use xlink:href="#quatroCasas" transform="translate(80)" />
      <use xlink:href="#quatroCasas" transform="translate(120)" />
    </g>

    <g id="tabuleiro">
      <use xlink:href="#fileiraDupla" />
      <use xlink:href="#fileiraDupla" transform="translate(0,40)" />
      <use xlink:href="#fileiraDupla" transform="translate(0,80)" />
      <use xlink:href="#fileiraDupla" transform="translate(0,120)" />
    </g>
  </defs>
  <use xlink:href="#tabuleiro" />
</svg>
```



28 linhas
~1400 chars



id e <use>

- **<use>** acessam outro elemento SVG por **referência**
 - Elementos que serão referenciados devem ter um **id**
 - O elemento referenciado pode estar no mesmo arquivo ou em outro, local ou remoto
 - **<use>** não muda atributos do elemento referenciado
- **Sintaxe**

```
<use xlink:href="uri"  
      x="coordinate"  
      y="coordinate"  
      width="length"  
      height="length"  
      style-attribute="..." (atribos fill, stroke, etc.)  
>
```
- **URI local**
 - `xlink:href="#name"`
- **URI externa**
 - `xlink:href="../../struct/Use01.svg/#name"`



Definições <defs>

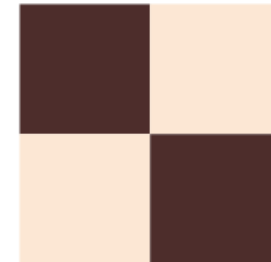
- Se você deseja reusar um recurso SVG, coloque em seção <defs>
 - O bloco <defs> armazena definições
 - Elementos declarados em <defs> não serão desenhados (para desenhá-lo, é preciso referenciá-lo com <use>
 - Use <defs> para guardar objetos que serão reusados com frequência
 - Ao reusar um objeto, pode-se mudar seu estilo e coordenadas

- Exemplo:

```
<svg ...>
  <defs>
    <rect id="preto" x="0" y="0" width="20" height="20" fill="rgb(64,32,32)" />
    <rect id="branco" x="0" y="0" width="20" height="20" fill="rgb(255,225,200)" />
  </defs>

  <use xlink:href="#preto" />
  <use xlink:href="#branco" transform="translate(20)" />
  <use xlink:href="#branco" transform="translate(0,20)" />
  <use xlink:href="#preto" transform="translate(20,20)" />

</svg>
```





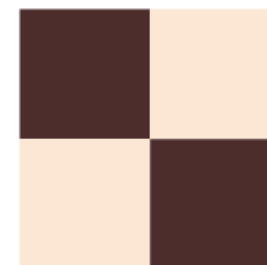
Grupos <g>

- **Grupos** permitem tratar um conjunto de objetos como um só
 - O grupo comporta-se como um objeto único
 - Elementos contidos no grupo herdam estilos e atributos comuns
 - Podem conter outros grupos

- **Sintaxe**

```
<g id="nome"  
  atributos de estilo  
  transform="comandos de transformação">  
  <!-- conteúdo do grupo -->  
</g>
```

```
<svg ...>  
  <defs>  
    ...  
    <g id="quatroCasas" transform="translate(10,10)">  
      <use xlink:href="#preto" />  
      <use xlink:href="#branco" transform="translate(20)"/>  
      <use xlink:href="#branco" transform="translate(0,20)"/>  
      <use xlink:href="#preto" transform="translate(20,20)"/>  
    </g>  
  </defs>  
  <use xlink:href="#quatroCasas" />  
</svg>
```





<title> e <desc>

- <title> e <desc> são elementos usados para documentar um SVG
 - Podem ser usados em **qualquer** elemento
 - O visualizador não precisa mostrá-los
 - Geralmente <title> aparece como *tooltip**; o mesmo pode acontecer com <desc>

```
<g id="tabuleiro">
  <title>Isto eh um tabuleiro</title>
  <desc>0 tabuleiro é construido da seguinte ...</desc>
  <use xlink:href="#fileiraDupla" />
  <use xlink:href="#fileiraDupla" transform="translate(0,40)" />
  <use xlink:href="#fileiraDupla" transform="translate(0,80)" />
  <use xlink:href="#fileiraDupla" transform="translate(0,120)" />
</g>
```

* A especificação sugere mas não obriga os visualizadores a implementar este comportamento



Símbolos <symbol>

- Um <symbol> é uma figura definida pelo usuário
 - Use para criar figuras complexas que serão reusadas, redimensionadas e reposicionadas
 - Símbolos podem conter figuras, outros símbolos, grupos
- É um elemento estrutural
 - Mais importante pelo que representa do que pelo que faz
 - Você também pode criar figuras com <g> (mas <g> deve ser usado para **agrupar** objetos)
 - Símbolos não são desenhados automaticamente (como acontece com um grupo, se estiver fora de um <defs>): é preciso referenciá-los (ex: via <use>)
- Símbolos podem ter um **viewBox**
 - A figura é geralmente desenhada dentro de um **viewBox**



Símbolos

```
<defs>
  <marker id="seta" ...> ... </marker>

  <symbol id="diodo" viewBox="0 0 10 10"
    stroke-opacity="0">
    <rect x="0" y="0" height="10"
      width="10" fill="white"/>
    <polygon points="0,0 10,0 5,8" />
    <rect x="0" y="8" width="10" height="2" />
  </symbol>
```

O mesmo pode-se fazer com `<g>` (exceto usar um `viewBox`), mas `<symbol>` é a melhor opção para este tipo de estrutura

```
<symbol id="led" viewBox="0 0 20 10">
  <use xlink:href="#diodo" x="5" y="0" height="10" width="10"/>
  <polyline points="15,3 18,5" stroke-width=".8" stroke="black"
    marker-end="url(#seta)"/>
  <polyline points="15,6 18,8" stroke-width=".8" stroke="black"
    marker-end="url(#seta)"/>
</symbol>
```

Reuso de um simbolo existente na criação de outro

```
</defs>
```

Uso do símbolo, com outra dimensão

```
<g transform="translate(70,25) rotate(-90,25,50)">
  <use xlink:href="#led" height="100" width="100"/>
</g>
<line x1="0" y1="50" x2="200" y2="50" stroke-width="4" stroke="black" />
```



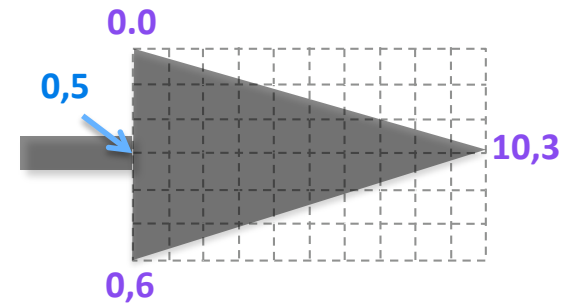
Marcadores <marker>

- Um marcador é um símbolo que **pode ser concatenado a vértices** de caminhos, linhas, polígonos e polilinhas
 - Geralmente usados para fazer pontas de setas ou similares
- Para criar
 - Defina um <marker> da mesma forma que um <symbol>
 - Use atributos **refX** e **refY** para marcar o ponto de conexão
 - Defina a orientação com **orient** (se for seta use **auto**)
 - Se **markerUnits** for **userSpaceOnUse**, use **markerHeight** e **markerWidth** para determinar as dimensões fixas do marcador
 - Se **markerUnits** for **strokeWidth**, o tamanho será proporcional à espessura da linha
- Para usar
 - <path>, <line>, <polyline> ou <polygon> têm atributos **marker-end**, **marker-start** ou **marker-mid**

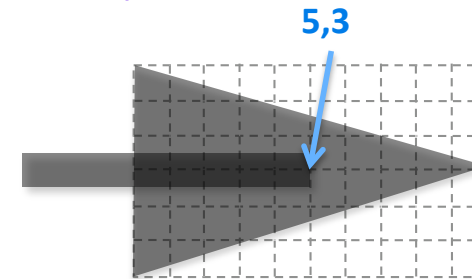


Marcadores

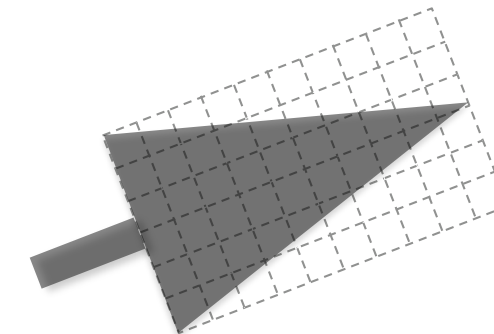
```
<marker viewBox="0 0 10 6"  
  refX="0" refY="5"  
  orient="0">  
  <polygon points="0,0 10,3 0,6" />  
</marker>
```



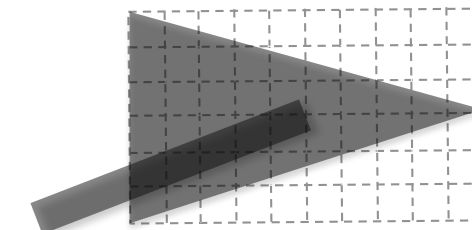
```
<marker viewBox="0 0 10 6"  
  refX="5" refY="3"  
  orient="auto">  
  <polygon points="0,0 10,3 0,6" />  
</marker>
```



```
<marker viewBox="0 0 10 6"  
  refX="0" refY="5"  
  orient="auto">  
  <polygon points="0,0 10,3 0,6" />  
</marker>
```



```
<marker viewBox="0 0 10 6"  
  refX="5" refY="3"  
  orient="0">  
  <polygon points="0,0 10,3 0,6" />  
</marker>
```

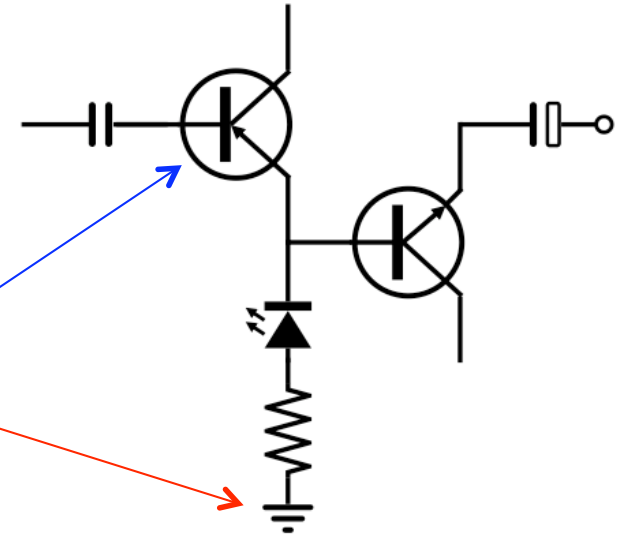




Marcadores e símbolos

```
<defs>
  <marker id="seta" viewBox="0 0 10 10" refX="5" refY="5"
    orient="auto" markerUnits="strokeWidth">
    <polyline points="0,0 10,5 0,10" />
  </marker>
  <marker id="terra" viewBox="0 0 10 6" refX="5" refY="0"
    orient="-90" markerUnits="userSpaceOnUse"
    markerHeight="18" markerWidth="24">
    <polyline points="1,1 9,1" .../>
    <polyline points="2.5,3 7.5,3" .../>
    <polyline points="4.5,5 5.5,5" .../>
  </marker>

  <g id="baseColetor"> ... </g>
  <symbol id="transistorPNP" viewBox="-1 -1 22 22">
    <use xlink:href="#baseColetor" />
    <line x1="0" x2="9" y1="20" y2="10" ... marker-end="url(#seta)" />
  </symbol>
  <marker id="setaPNP" viewBox="0 0 10 10" refX="0" refY="5" orient="auto"
    markerUnits="userSpaceOnUse" markerWidth="50" markerHeight="50">
    <use xlink:href="#transistorPNP" x="0" y="0" height="10" width="10" transform="rotate(-90,5,5)" />
  </marker>
  ...
</defs>
<g id="circuito1" stroke-width="2" stroke="black" stroke-linecap="square" transform="translate(0,25)">
  <g transform="translate(50,0)">
    <polyline points="0,50 25,50" marker-end="url(#setaPNP)"/>
    <polyline points="72,26 72,0" />
    <polyline points="72,73 72,100"/>
  </g>
  <g transform="translate(122,125) rotate(90,0,25)">
    <polyline points="0,25 30,25 60,25" marker-mid="url(#setaRes)" marker-end="url(#terra)"/>
  </g>
  ...
</g>
```





Transparência e visibilidade

- Há três formas de variar a visibilidade de um objeto em SVG
 - Atributo **opacity**: mais abrangente que fill-opacity e stroke-opacity – se aplicado a um grupo, afeta todos os objetos; com opacity="0" todo o grupo fica invisível
 - Atributo **visibility**: pode ser **visible** ou **hidden**; se hidden o objeto fica invisível
 - Atributo **display="none"**: não apenas torna o objeto invisível, como remove-o da árvore (não é possível capturar eventos)
- A visibilidade pode ser controlada dinamicamente através de scripts
 - Todas as propriedades podem ser manipuladas via DOM (com JavaScript) e animações
- Símbolos, e quaisquer definições de grupos ou elementos em **<defs>** não existem (e são, portanto, invisíveis) enquanto não forem referenciados com **<use>**



Links

- SVG permite links internos e externos através de **IRIs** (Internationalized Resource Identifier*)
- Referências locais podem ser especificadas diretamente em atributos do XLink **xlink:href**
 - `#referencia`
- Em atributos dos elementos SVG é preciso chamar a referência através de uma função IRI
 - `url(#referencia)`
- Referências também podem ser feitas a arquivos externos
 - `http://w.com/outro.svg#referencia`
 - `url(http://w.com/outro.svg#referencia)`
- Elementos **<a>** permitem linkar para fora do SVG
 - `<a xlink:href="http://www.site.com/a.html"> <rect...> `



Exercícios

- Adapte os desenhos feitos no exercício anterior para uma estrutura usando
 - grupos e símbolos
 - defs
 - use e id
- Construa um componente formado por retângulos, círculos, etc. e associe a um símbolo
 - Depois use esse grupo para construir outros objetos
 - Armazene as partes em arquivos separados