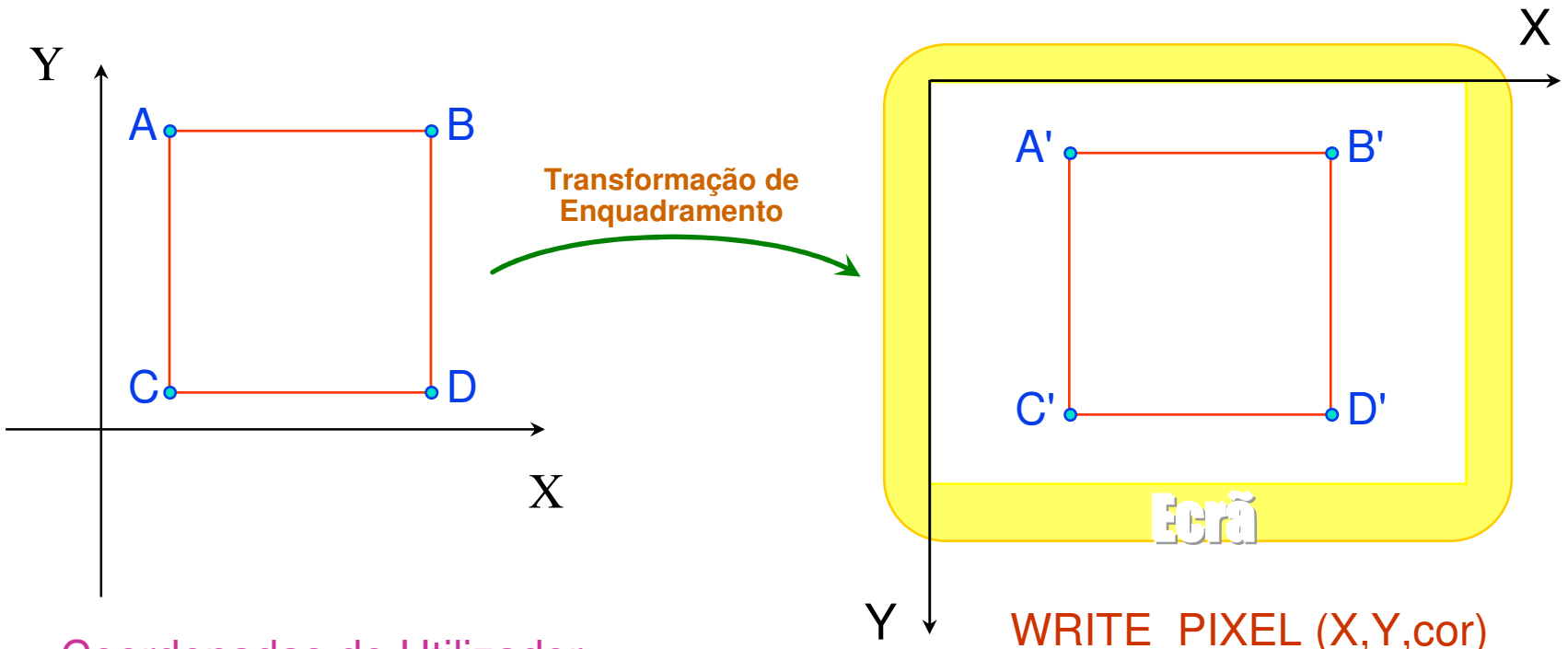
Enquadramento



Coordenadas do Utilizador

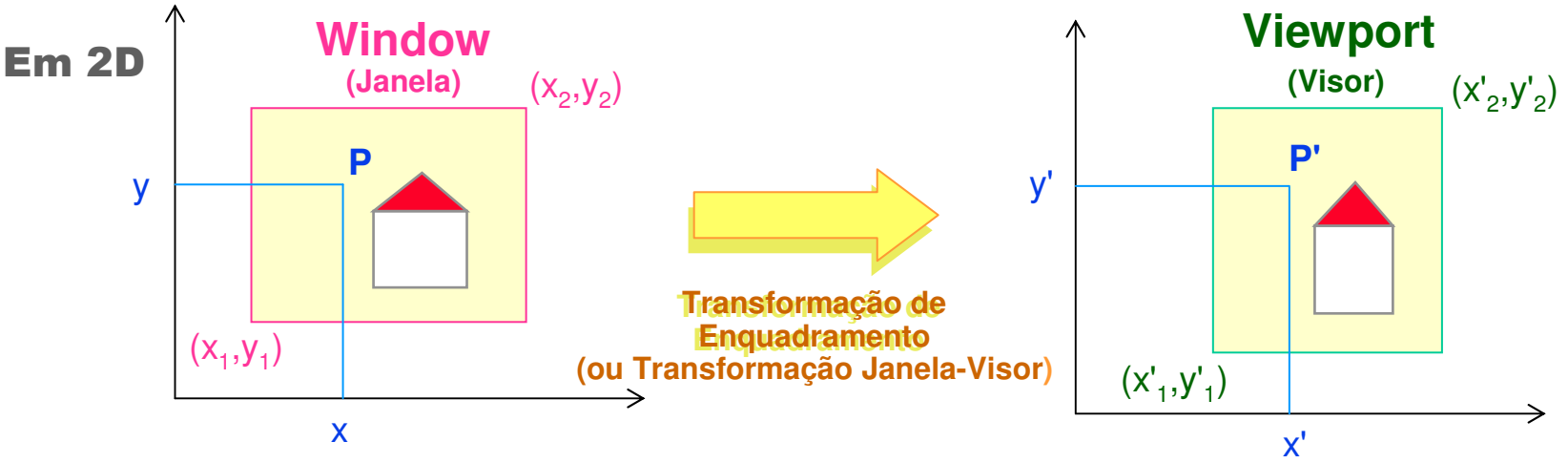
(*World Coordinates* ≡ **WC**)

WRITE_PIXEL (X,Y,cor)

Coordenadas do Dispositivo

(*Device Coordinates* ≡ **DC**)

Enquadramento



Coordenadas do utilizador

Coordenadas do dispositivo físico
(e.g. impressora)

$$\frac{x_2 - x_1}{x - x_1} = \frac{x'_2 - x'_1}{x' - x'_1} \Rightarrow x' = (x - x_1)\,(x'_2 - x'_1) / (x_2 - x_1) + x'_1$$

$$x' = (x - \mathbf{A}) \cdot \mathbf{B} + \mathbf{C}$$

$$\frac{y_2 - y_1}{y - y_1} = \frac{y'_2 - y'_1}{y' - y'_1} \Rightarrow y' = (y - y_1)\,(y'_2 - y'_1) / (y_2 - y_1) + y'_1$$

$$y' = (y - \mathbf{D}) \cdot \mathbf{E} + \mathbf{F}$$

Pseudo-código da especificação: janela(x1, x2, y1, y2) visor(x1', x2', y1', y2')

M.Próspero

Formato ou **Relação de Aspecto** (***Aspect Ratio***) de um retângulo:

$$a = \frac{x_{max} - x_{min}}{y_{max} - y_{min}}$$

Exemplos bem conhecidos

4:3

16:9

Condição para manter as proporções da imagem no enquadramento:

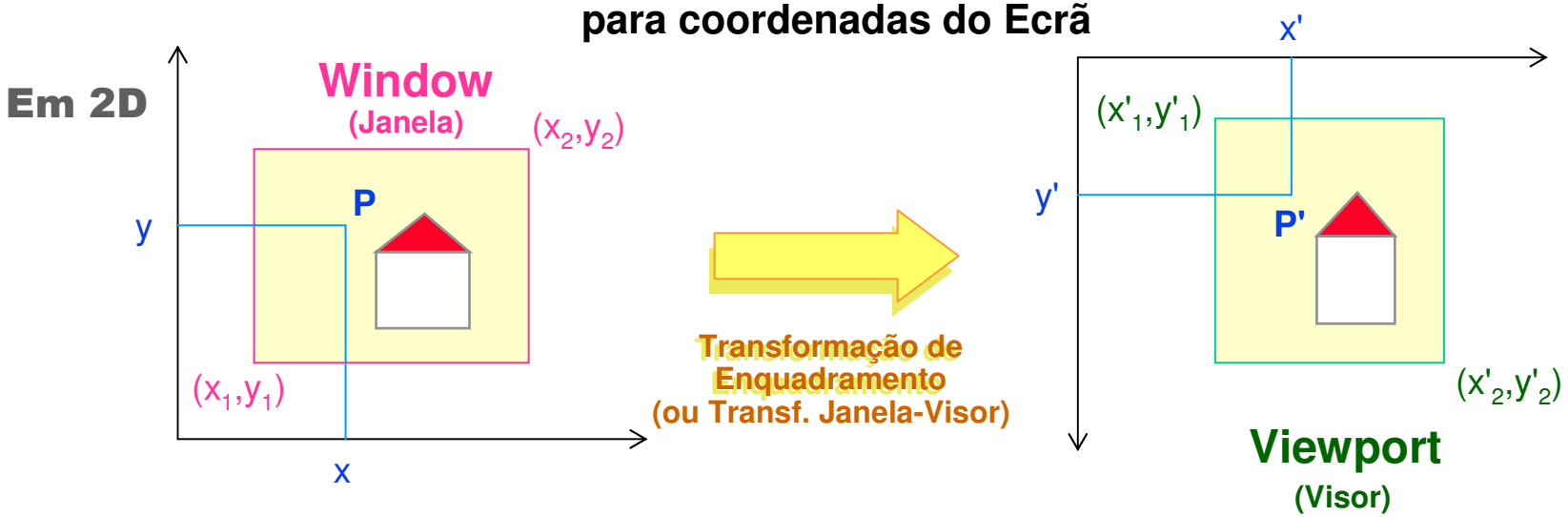
$a_{janela} = a_{visor}$ ⇒ $\frac{x_2 - x_1}{y_2 - y_1} = \frac{x'_2 - x'_1}{y'_2 - y'_1}$ ⇒ **E = B**

Caso contrário ⇒ **distorção da imagem**

M.Próspero

Enquadramento

para coordenadas do Ecrã



$$\frac{x_2 - x_1}{x - x_1} = \frac{x'_2 - x'_1}{x' - x'_1} \Rightarrow x' = (x - x_1)\,(x'_2 - x'_1) / (x_2 - x_1) + x'_1$$

$$x' = (x - \mathbf{A}) \,.\, \mathbf{B} + \mathbf{C}$$

$$\frac{y_2 - y_1}{y - y_1} = \frac{y'_2 - y'_1}{y'_2 - y'} \Rightarrow y' = - (y - y_1)\,(y'_2 - y'_1) / (y_2 - y_1) + y'_2$$

$$y' = (y - \mathbf{D}) \,.\, \mathbf{G} + \mathbf{H}$$

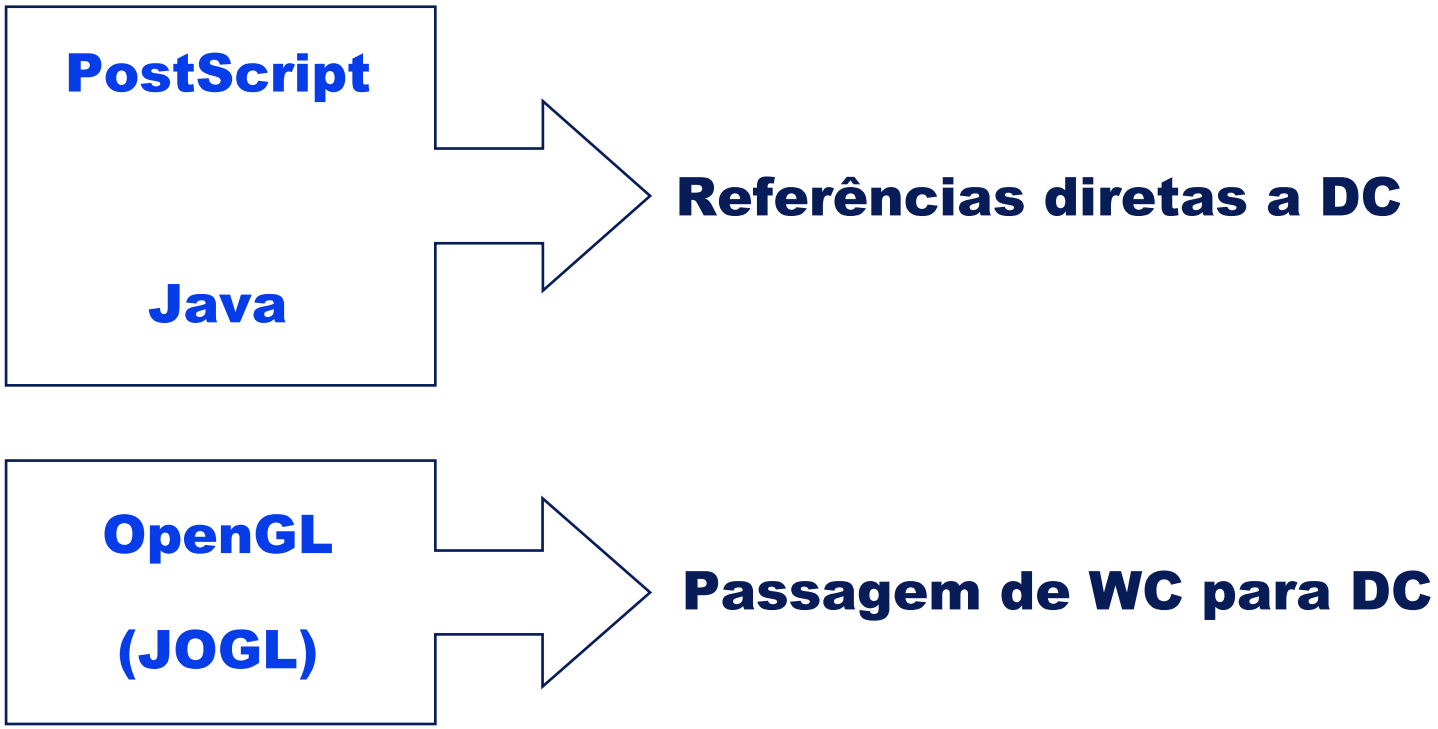
$$\mathbf{G} = - \mathbf{E}$$

Pseudo-código da especificação: janela(x1, x2, y1, y2) visor(x1', x2', y1', y2')

M.Próspero

Enquadramento

Ver os exemplos apresentados anteriormente nas aulas:



Especificação OpenGL de uma JANELA:

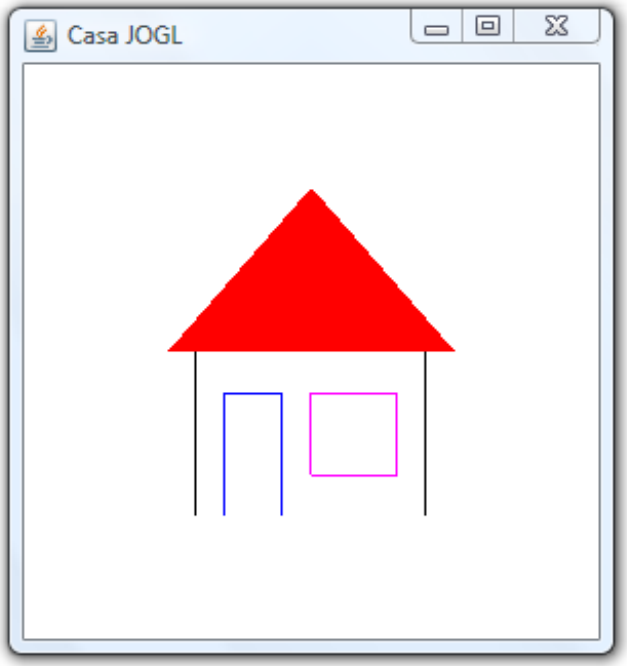
```
gluOrtho2D (0.0, 400.0, -280.0, 0.0);
```

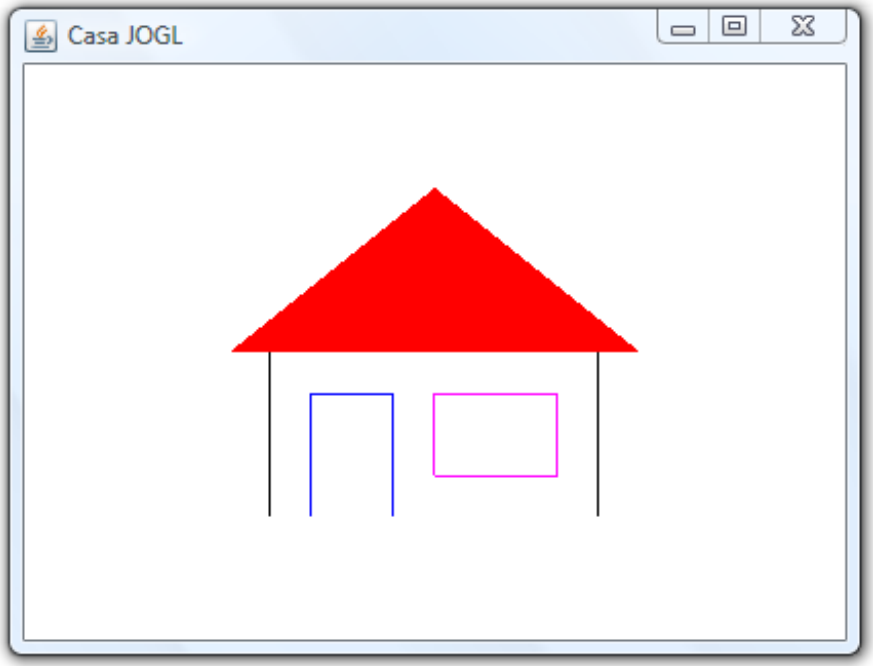
Em WC

Especificação OpenGL de um VISOR:

```
glViewport (0, 0, 400, 280);
```

Em pixels, mas não exatamente em DC






```
janela(0, 400, -280, 0)
visor(0, 280, 0, 280)   ⟶  a_v = 1:1
```

Pseudo-código, em WC

Pseudo-código, em DC

```
janela(0, 400, -280, 0)
visor(0, 400, 0, 280)
```

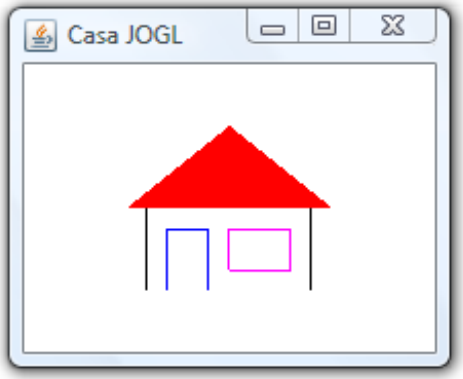
a = 10:7

$a_j = a_v$

$a_j \neq a_v$

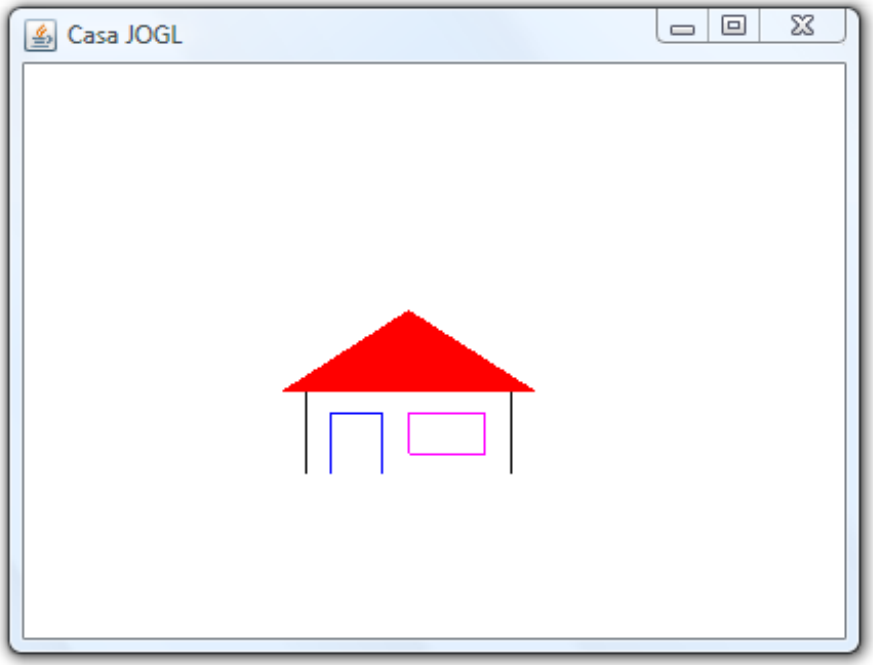
distorção da imagem

M.Próspero




```
janela(0, 400, -280, 0)
visor(0, 200, 0, 140)
```

} a = 10:7




```
janela(-100, 540, -380, 180)
visor(0, 400, 0, 280)
```

→ a_j = 8:7

⇓

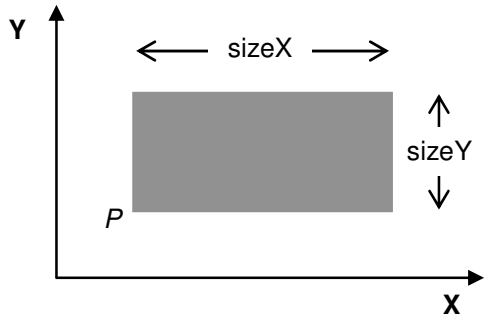
distorção da imagem

ESPECIFICAÇÃO DO VISOR NA ÁREA DE DESENHO (1)

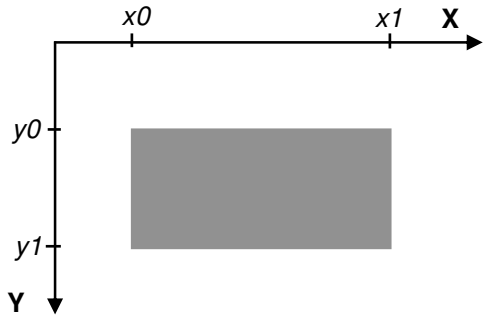
void glViewport(GLint *x*, GLint *y*, GLsizei *width*, GLsizei *height*);

Defines a pixel rectangle in the window into which the final image is mapped. The (*x*, *y*) parameter specifies the lower-left corner of the viewport, and *width* and *height* are the size of the viewport rectangle. By default, the initial viewport values are (*0*, *0*, *winWidth*, *winHeight*), where *winWidth* and *winHeight* are the size of the window.

draw area, not WC



glViewport(*Px*, *Py*, sizeX, sizeY)

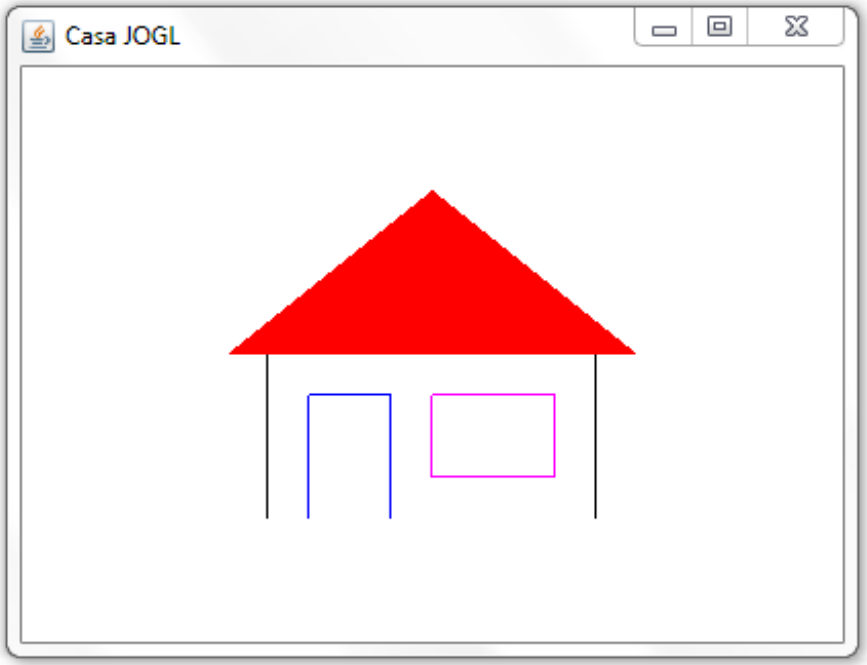


DC num ecrã

visor(*x0*, *x1*, *y0*, *y1*)

ESPECIFICAÇÃO DO VISOR NA ÁREA DE DESENHO (2)

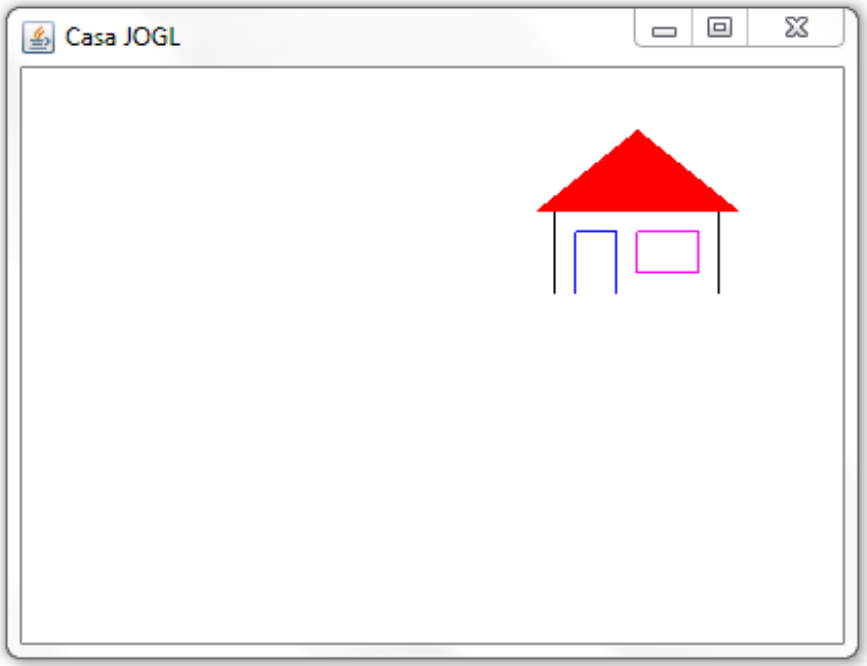
EXEMPLOS numa área de desenho (ou canvas) com 400x280 pixels




```
glViewport (0, 0, 400, 280)
```

OU

```
visor(0, 400, 0, 280)
```




```
glViewport (200, 140, 200, 140)
```

OU

```
visor(200, 400, 0, 140)
```

Cálculo matemático por transformação linear?

Será possível que a transformação de enquadramento possa ser calculada por uma matriz M que, aplicada a um ponto P da janela (WC) o converta num ponto P' do visor (DC)?

Se se escrever P na forma do seguinte vetor

$$P = \begin{bmatrix} x \\ y \end{bmatrix}$$

então a operação de transformação teria de ser

$$P' = M.P$$

Mas não existe nenhuma matriz 2x2 que satisfaça a operação!

A solução está no uso de **coordenadas homogéneas**, passando M a ser do tipo 3x3.

Enquadramento janela-visor como transformação linear

$$x' = (x - \mathbf{A}) . \mathbf{B} + \mathbf{C} = x.\mathbf{B} - \mathbf{A}.\mathbf{B} + \mathbf{C}$$

$$y' = (y - \mathbf{D}) . \mathbf{J} + \mathbf{K} = y.\mathbf{J} - \mathbf{D}.\mathbf{J} + \mathbf{K}$$

J = **E** *ou* **G** **K** = **F** *ou* **H**

P' = M.P

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} B & 0 & C-A.B \\ 0 & J & K-D.J \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Pontos **P** e **P'** necessariamente em coordenadas homogéneas!

Obs.: Mais tarde se apresentará esta matriz como composição de transformações geométricas.

M.Próspero