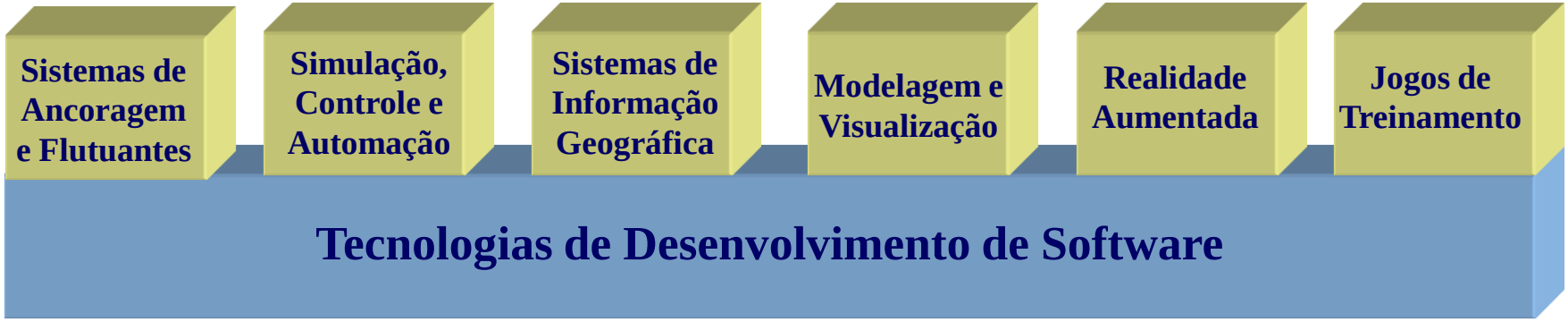
Ferramentas de Desenvolvimento do Tecgraf: IUP, CD e IM

Antonio Scuri

scuri@tecgraf.puc-rio.br

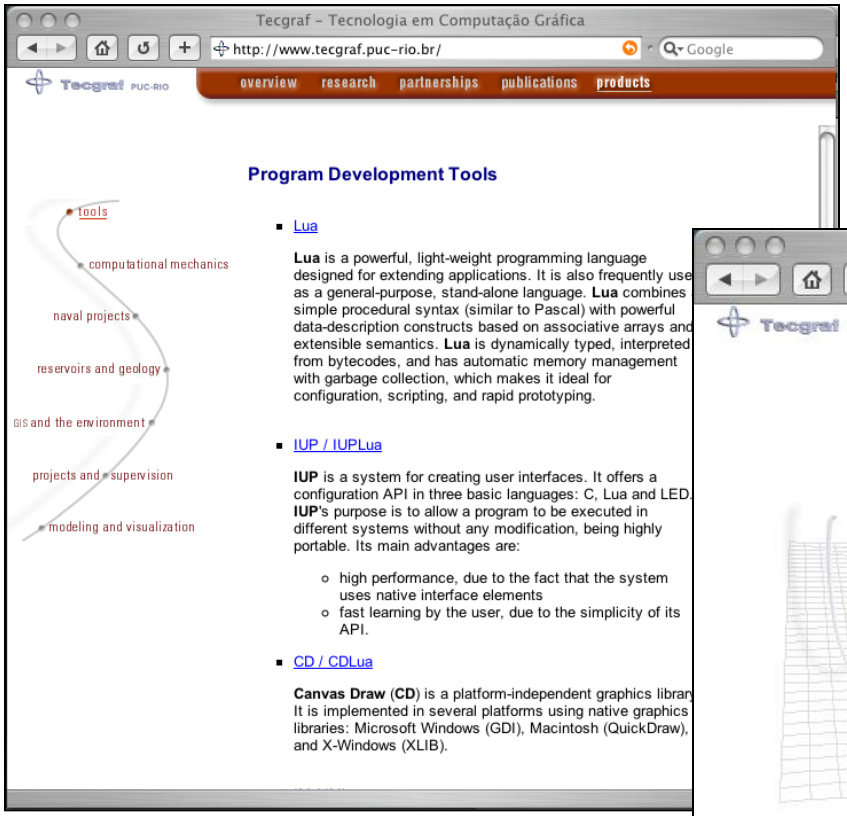
O Laboratório

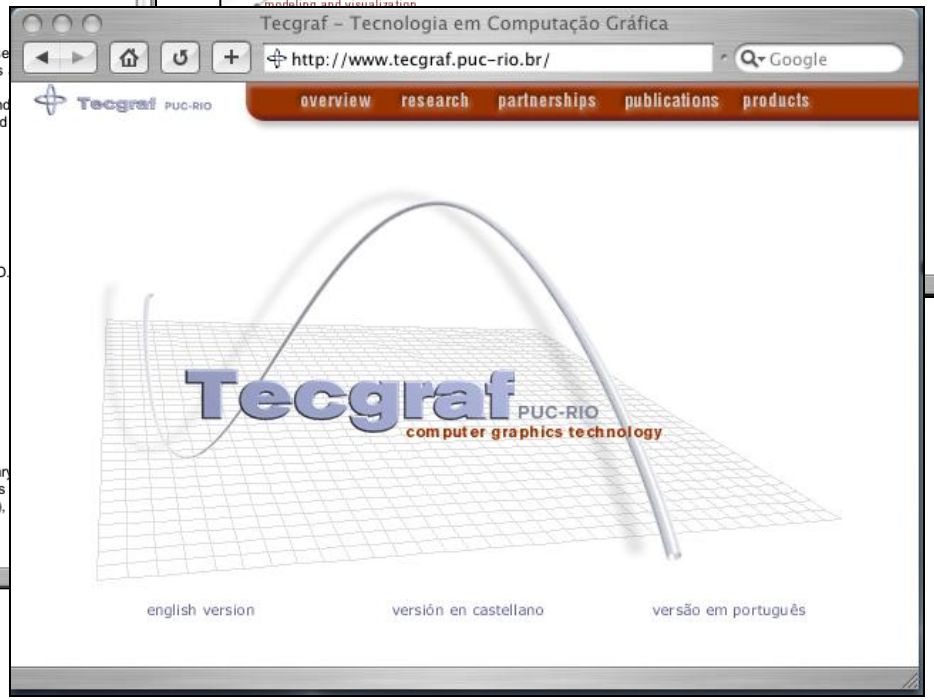
Tecgraf
PUC-RIO



Para mais Informações

http://www.tecgraf.puc-rio.br/





As Ferramentas

http://www.tecgraf.puc-rio.br/cd/

http://www.tecgraf.puc-rio.br/iup/

http://www.tecgraf.puc-rio.br/im/

Webbook
Tecmake
LuaBinaries

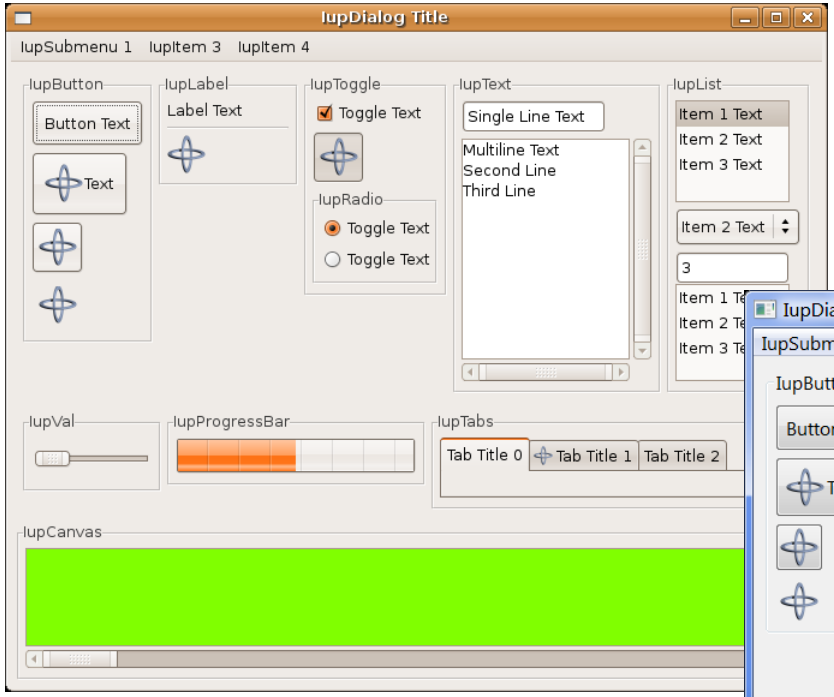
Características Gerais

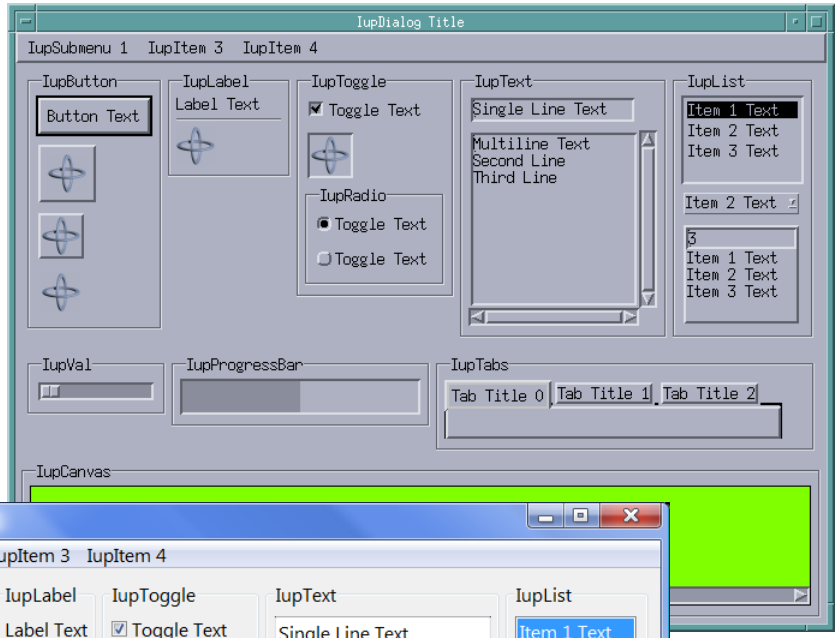
- Modelo de Abstração
- Portabilidade (Desktop)
- Desempenho (recursos nativos do sistema)
- Simplicidade com Recursos
- Livre e Código Aberto (mesma licença de Lua)
- Binários de várias plataformas prontos para Download
- API em C e em Lua
- Extensa Documentação com Exemplos
- Política de Compatibilidade entre Versões
- Melhor Curva de Aprendizado (*)

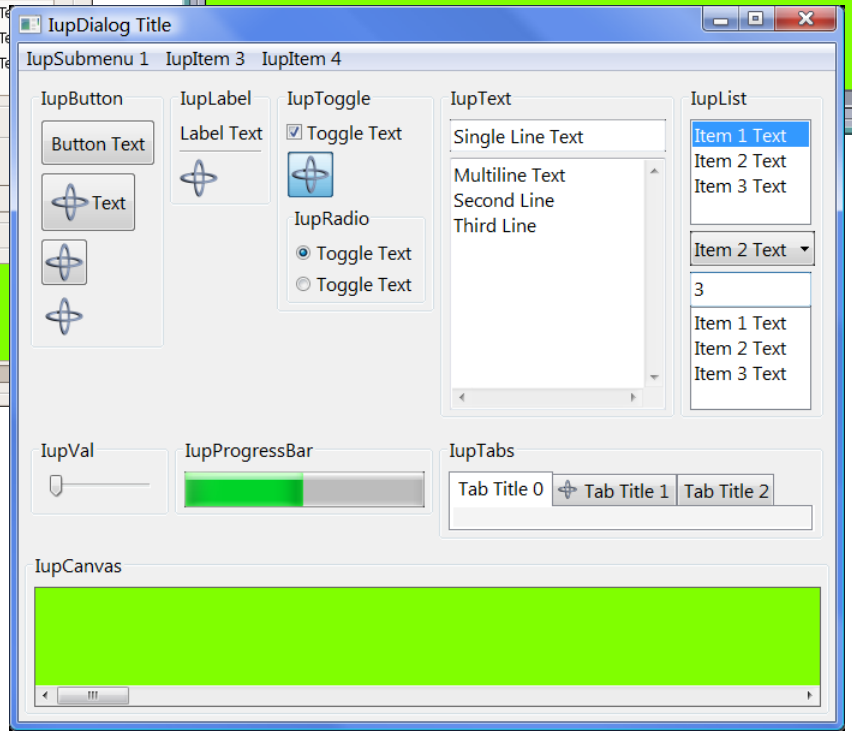
IUP - Interface com o Usuário Portável

Motif em MWM

GTK em Gnome

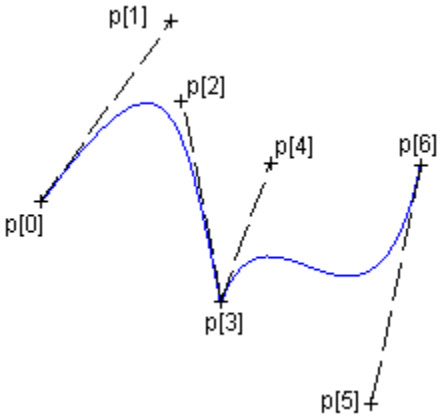


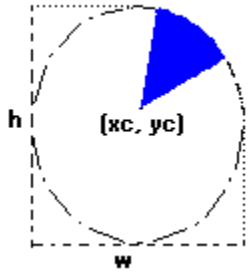


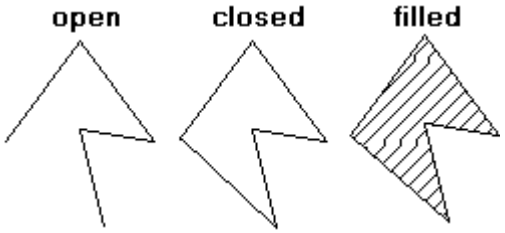


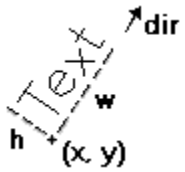
Windows Vista

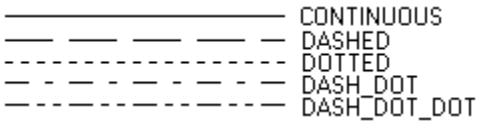
CD – Biblioteca Gráfica 2D

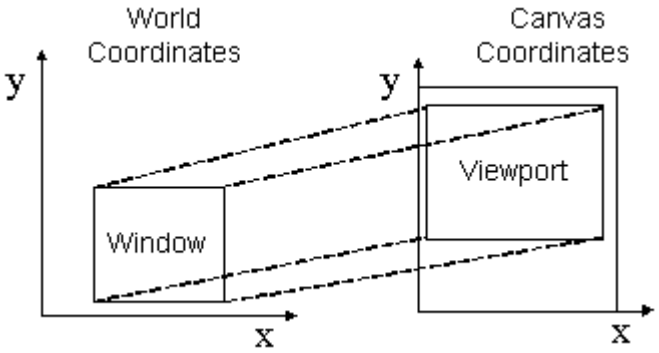


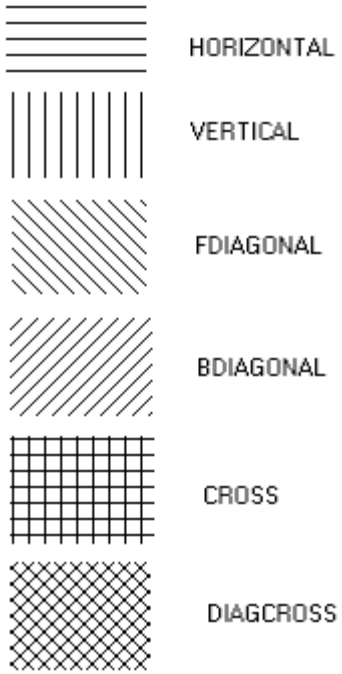




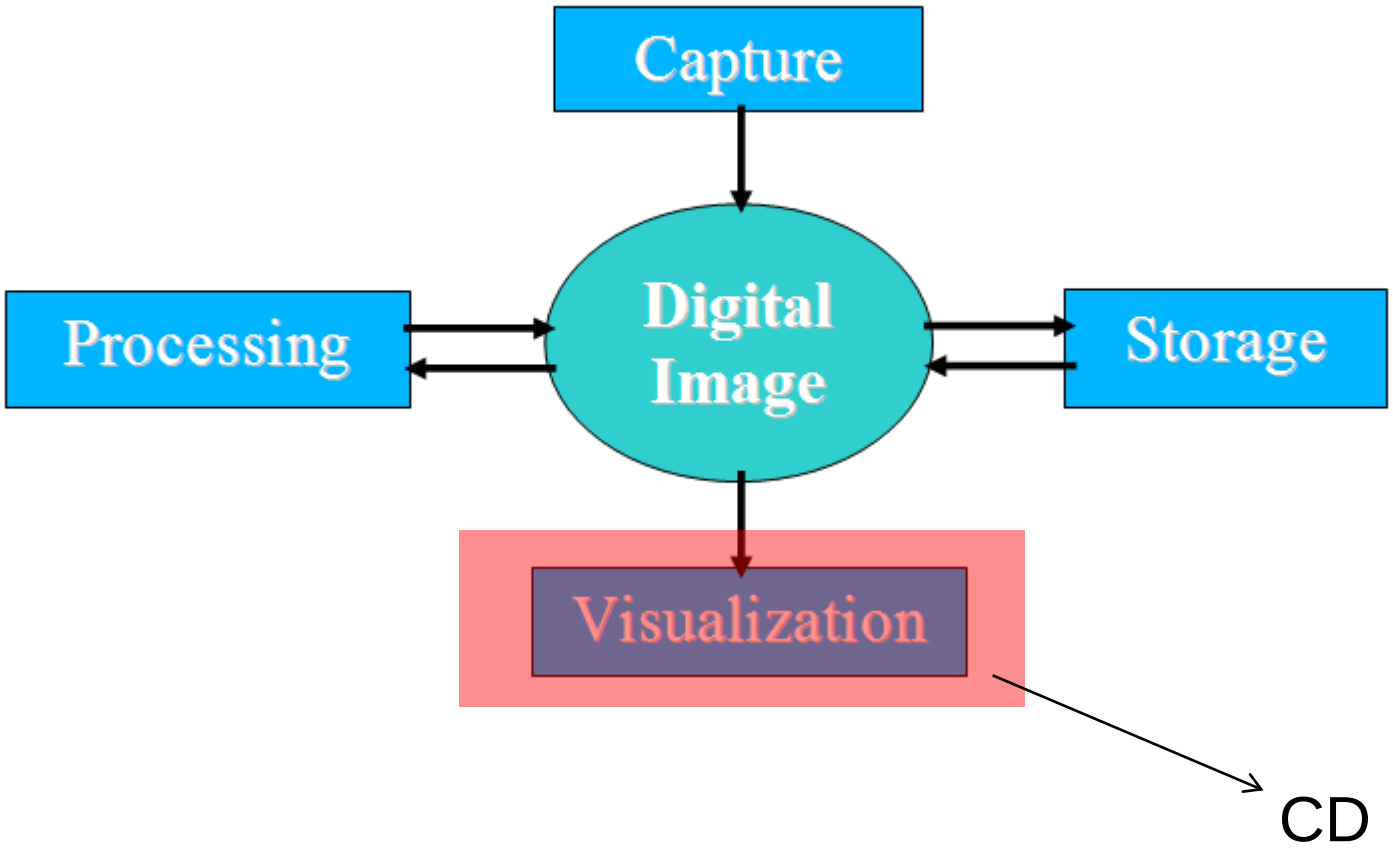




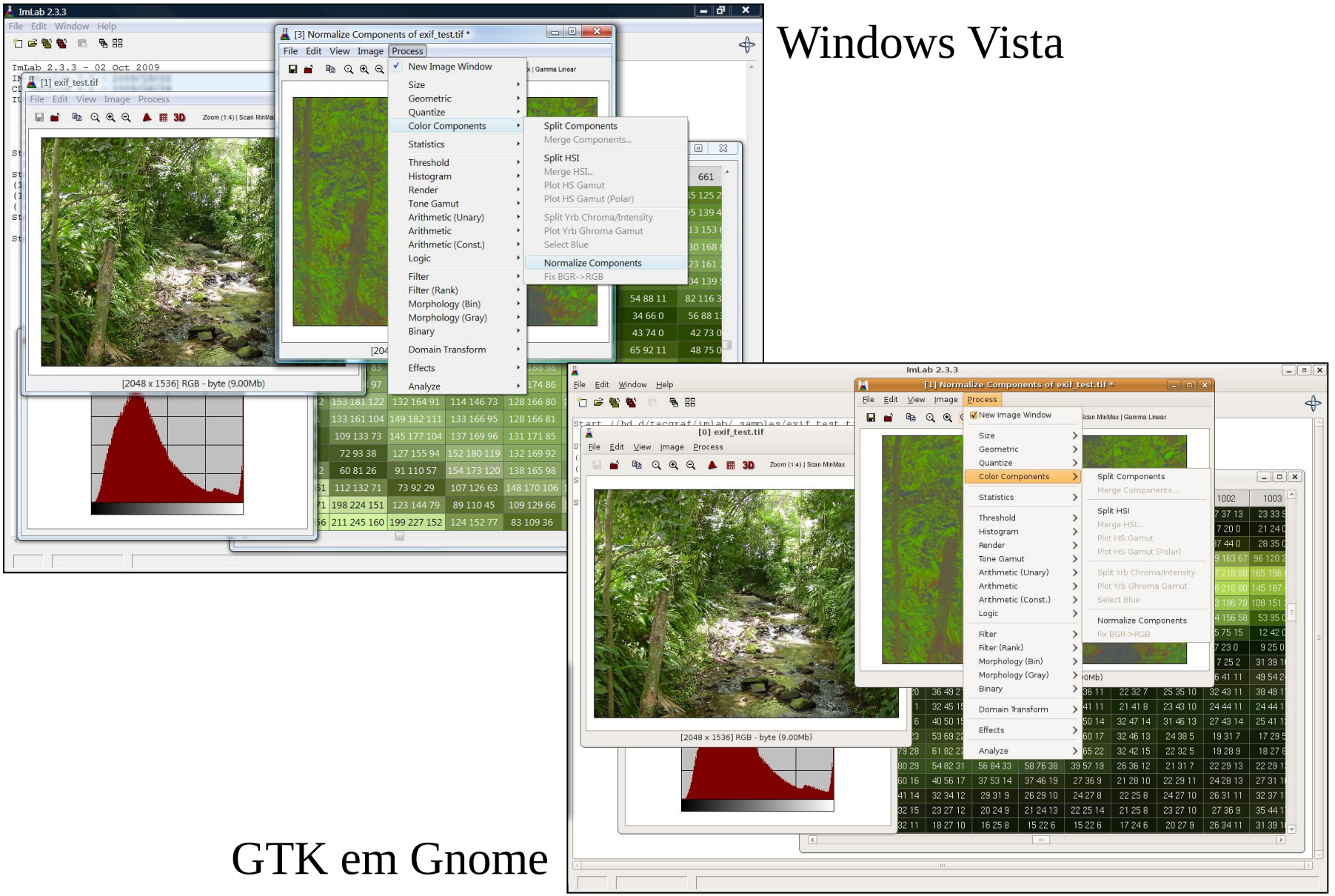




IM – Representação, Armazenamento, Captura, Processamento e Análise de Imagens



Exemplo de uma Aplicação Completa



Windows Vista

GTK em Gnome

Características Diferenciais

- **IUP**: layout abstrato. Construção do diálogo sem o posicionamento manual dos controles.
- **CD**: mesma API para diversos tipos de Canvas (Janela, Impressora, Clipboard, Imagem, Postscript)
- **IM**: foco em aplicações científicas mas flexibilidade para atender quaisquer aplicações.

Binding Lua

- Inicialização
  ```
  require"imlua"
  require"cdlua"
  require"iuplua"
  ```
  (LUA_CPATH)
- Regras gerais
  - Funções: `imXxx => im.Xxx`
  - Definições: `IM_XXX => im.XXX`
  - Métodos: `imFileXXX(ifile,... => ifile:XXX(...`
- Construtores especiais para IUP

Binding Lua – Controles IUP

- Metatable:
  - `IupSetAttribute(label, "TITLE", "test")  =>`
    `label.title = "test"`
  - `title = IupGetAttribute(label, "TITLE")  =>`
    `title = label.title`
  - `IupSetCallback(button, "ACTION", button_action_cb);  =>`
    `function button:action() ... end`
- Construtores:
  - `IupButton("test")  =>`
    `iup.button{title = "test", alignment="acenter"}`
  - `IupHbox(bt1, bt2, NULL)  =>`
    `iup.hbox{bt1, bt2, margin="10x10"}`

Exemplos IM - Indexação

```
require"imlua"

local filename = "lena.jpg"
local image = im.FileImageLoad(filename)

local r = image[0]
local g = image[1]
local b = image[2]

for row = 0, image:Height() - 1, 10 do
   for column = 0, image:Width() - 1, 10 do
        r[row][column] = 0
        g[row][column] = 0
        b[row][column] = 0
   end
end

image:Save("lena_indexing.bmp", "BMP")
```

Exemplos IM - Processamento

```
image, err = im.FileImageLoad(file_name);

if (err and err ~= im.ERR_NONE) then
  error(err_msg[err+1])
end

local new_image = im.ProcessCropNew(image, x1, x2, y1, y2)
image:Destroy()

new_image:Save(file_name, "JPEG");
new_image:Destroy()
```

Exemplos IM - Análise

```
local image = im.FileImageLoad(filename)

local gray = im.ImageCreate(image:Width(), image:Height(), im.GRAY,
   image:DataType())
im.ConvertColorSpace(image, gray)
gray:Save("lena_gray.jpg", "JPEG")

local binary = im.ImageCreate(image:Width(), image:Height(), im.BINARY,
   image:DataType())
im.ProcessSliceThreshold(gray, binary, 0, 128)
binary:Save("lena_binary.jpg", "JPEG")

local region = im.ImageCreate(image:Width(), image:Height(), im.GRAY,
   im.USHORT)
local count = im.AnalyzeFindRegions(binary, region, 4, 1)
print("regions: ", count)
```

Exemplos CD - Postscript

```
local canvas = cd.CreateCanvas(cd.PS, "cdtest.ps")

canvas:MarkSize(40)
canvas:Mark(x, y)

canvas:Font("Courier", cd.PLAIN, 12)
canvas:TextAlignment(cd.CENTER)
canvas:Text(x, y, text)

local xmin, xmax, ymin, ymax = canvas:GetTextBox(x, y, text)
canvas:Rect(xmin, xmax, ymin, ymax)

canvas:Kill()
```

Exemplos CD com IM

```
require"imlua"
require"cdlua"
require"cdluaim"

local image = im.ImageCreate(500, 500, im.RGB, im.BYTE)
local canvas = image:cdCreateCanvas()  -- Creates a CD_IMAGERGB canvas

canvas:Activate()
canvas:Background(cd.EncodeColor(255, 255, 255))
canvas:Clear()
fgcolor = cd.EncodeColor(255, 0, 0) -- red
fgcolor = cd.EncodeAlpha(fgcolor, 50) -- semi transparent
canvas:Foreground(fgcolor)
canvas:Font("Times", cd.BOLD, 24)
canvas:Text(100, 100, "Test")
canvas:Line(0,0,100,100)
canvas:Kill()

image:Save("new.bmp", "BMP")
```

Exemplos IUP – Layout Abstrato

```
dlg = iup.dialog
{
  iup.hbox
  {
    iup.fill{},
    iup.button{title="Ok",size="40"},
    iup.button{title="Cancel",size="40"},
    iup.fill{}
   ;margin="10x10", gap="10"
  }
 ;title="Test"
}

dlg:show()
```

Callbacks em Lua

```
btok = dlg[1][2]
btcancel = dlg[1][3]

btok.bgcolor = "255 0 0"

function btok:action()
  local aux = self.fgcolor
  self.fgcolor = self.bgcolor
  self.bgcolor = aux
End

function btcancel:action()
  return iup.CLOSE
end
```

Exemplo com IUP, CD e IM

```
image = im.FileImageLoad("flower.jpg")
cnv = iup.canvas{rastersize = image:Width().."x"..image:Height(), border
   = "NO"}
cnv.image = image -- store the new image in the IUP canvas as an
   attribute

function cnv:map_cb()
  -- the CD canvas can only be created when the IUP canvas is mapped
  self.canvas = cd.CreateCanvas(cd.IUP, self)
end

function cnv:action()
  -- called everytime the IUP canvas needs to be repainted
  self.canvas:Activate()
  self.canvas:Clear()
  self.image:cdCanvasPutImageRect(self.canvas, 0, 0, 0, 0, 0, 0, 0, 0)
end

dlg = iup.dialog{cnv}
dlg:show()
```

Antonio Scuri
scuri@tecgraf.puc-rio.br