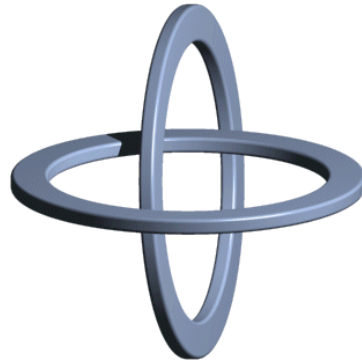


Ferramentas de Desenvolvimento do Tecgraf: IUP, CD e IM

Antonio Scuri

scuri@tecgraf.puc-rio.br





Tecgraf
PUC-RIO

**Sistemas de
Ancoragem
e Flutuantes**

**Simulação,
Controle e
Automação**

**Sistemas de
Informação
Geográfica**

**Modelagem e
Visualização**

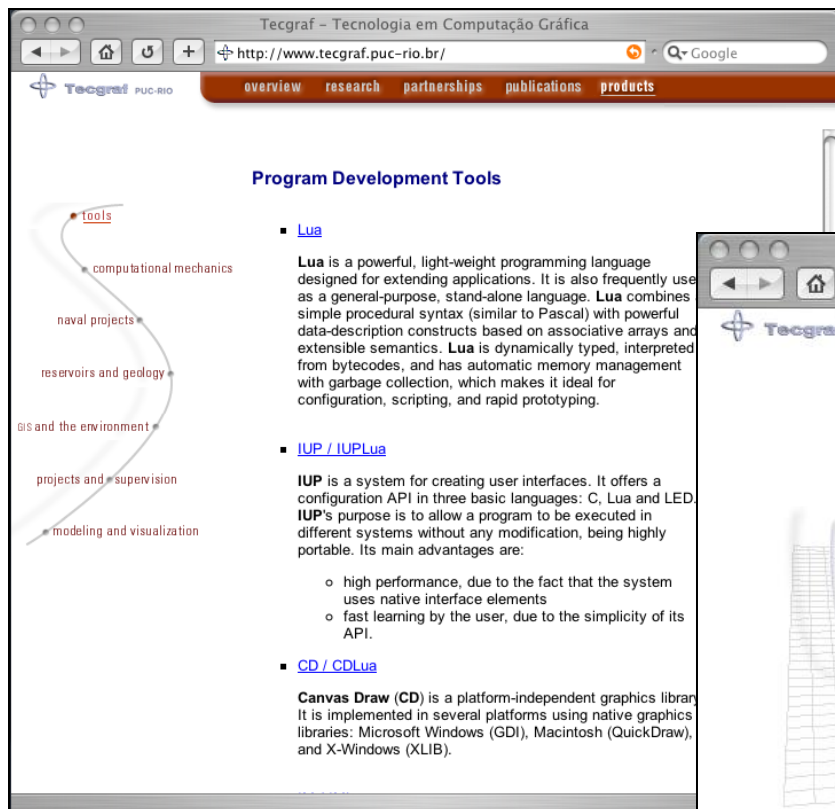
**Realidade
Aumentada**

**Jogos de
Treinamento**

Tecnologias de Desenvolvimento de Software

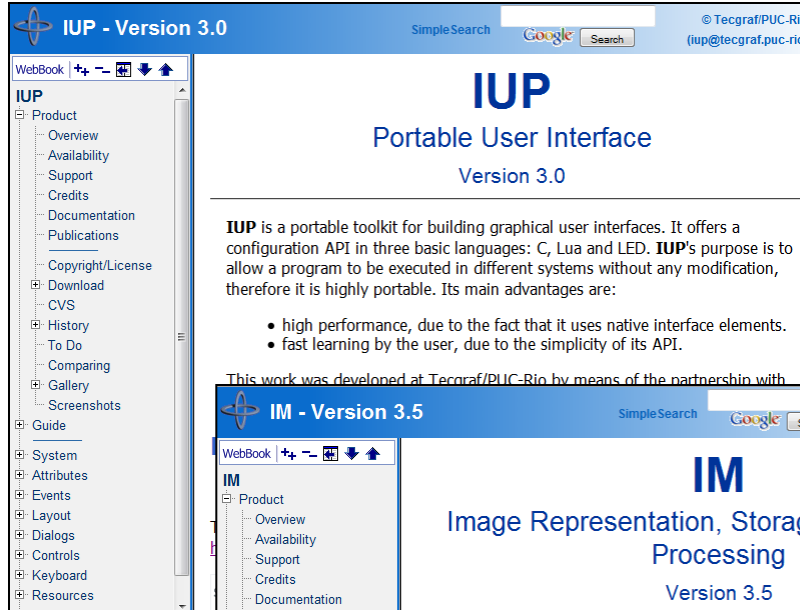
Para mais Informações

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

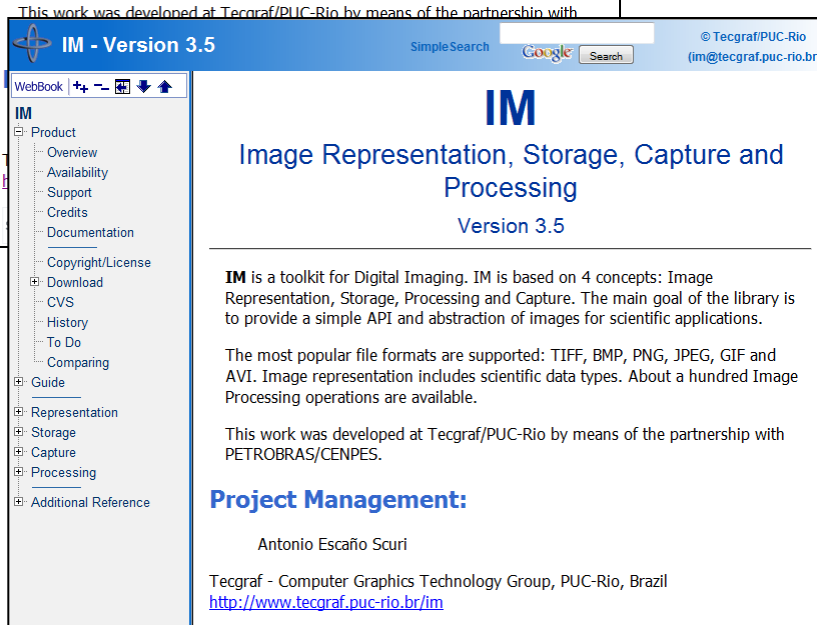
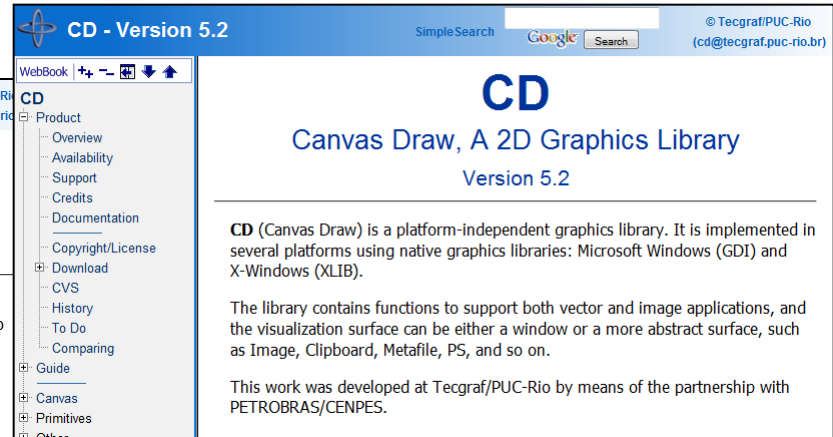


As Ferramentas

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



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



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

Webbook
Tecmake
LuaBinaries

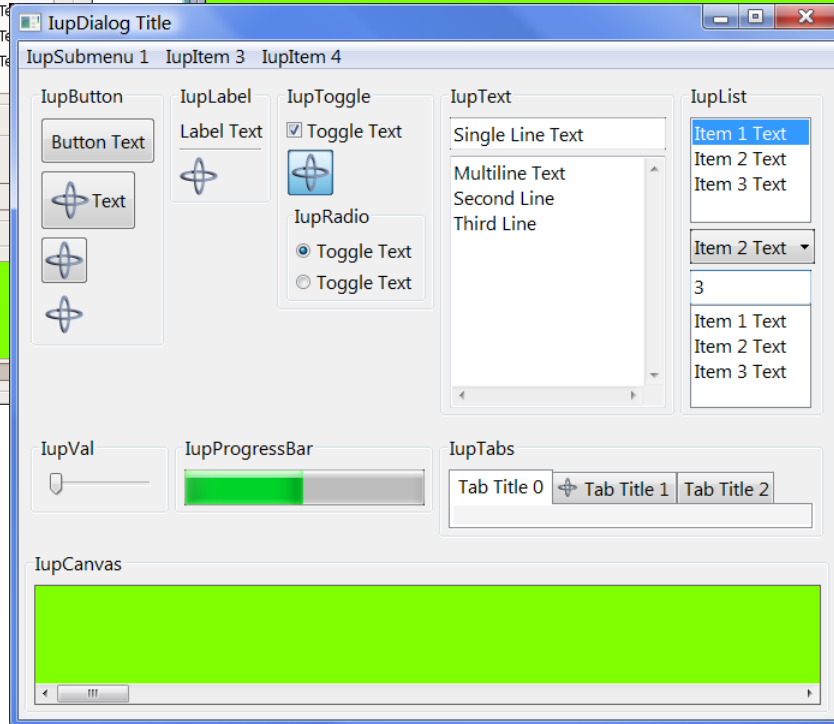
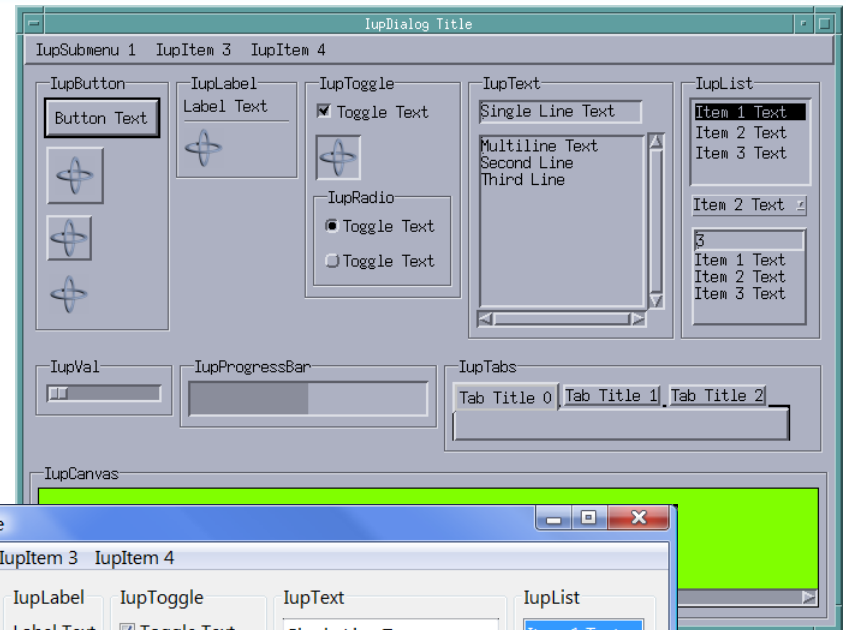
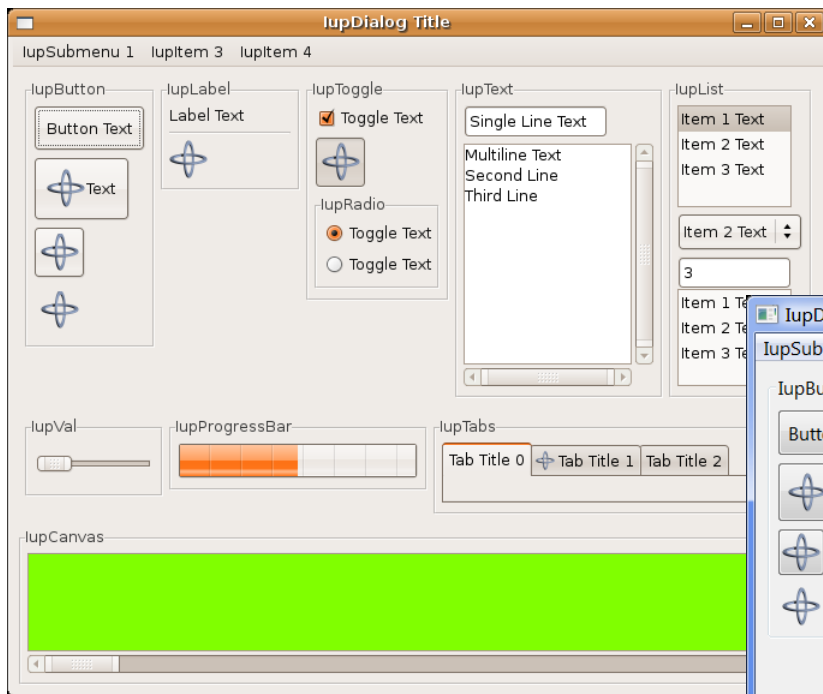


- Modelo de Abstração
- Portabilidade (Desktop)
- Desempenho (recursos nativos do sistema)
- Simplicidade com Recursos
- Livre e Código Aberto (mesma licença de Lua)
- Binários de várias plataformas prontos para Download
- API em C e em Lua
- Extensa Documentação com Exemplos
- Política de Compatibilidade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

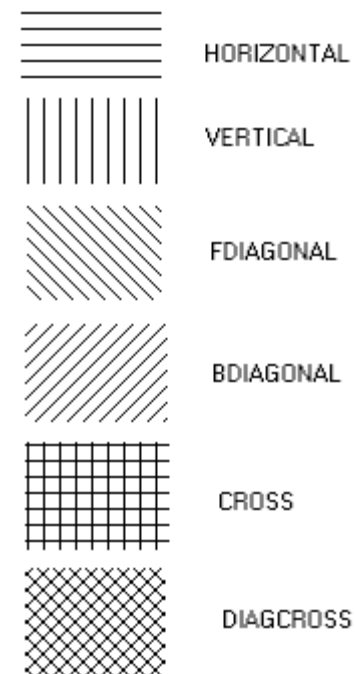
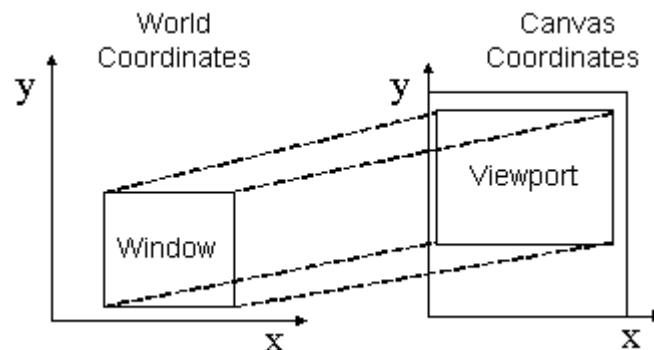
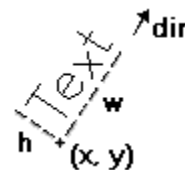
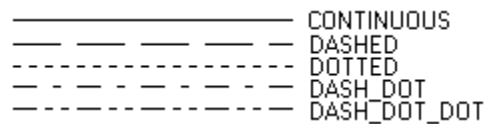
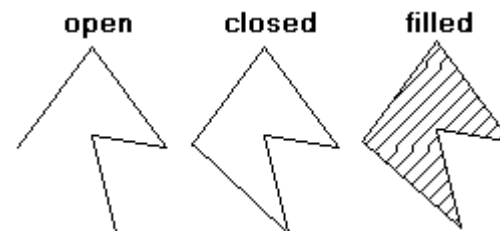
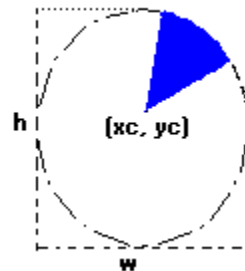
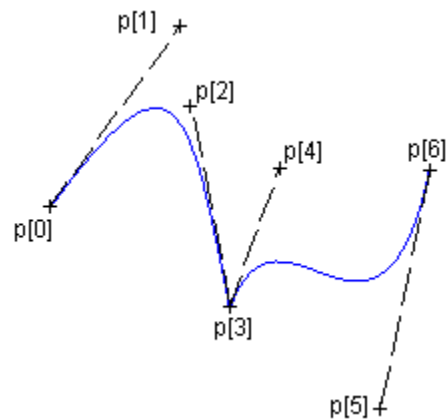
IUP - Interface com o Usuário Portável

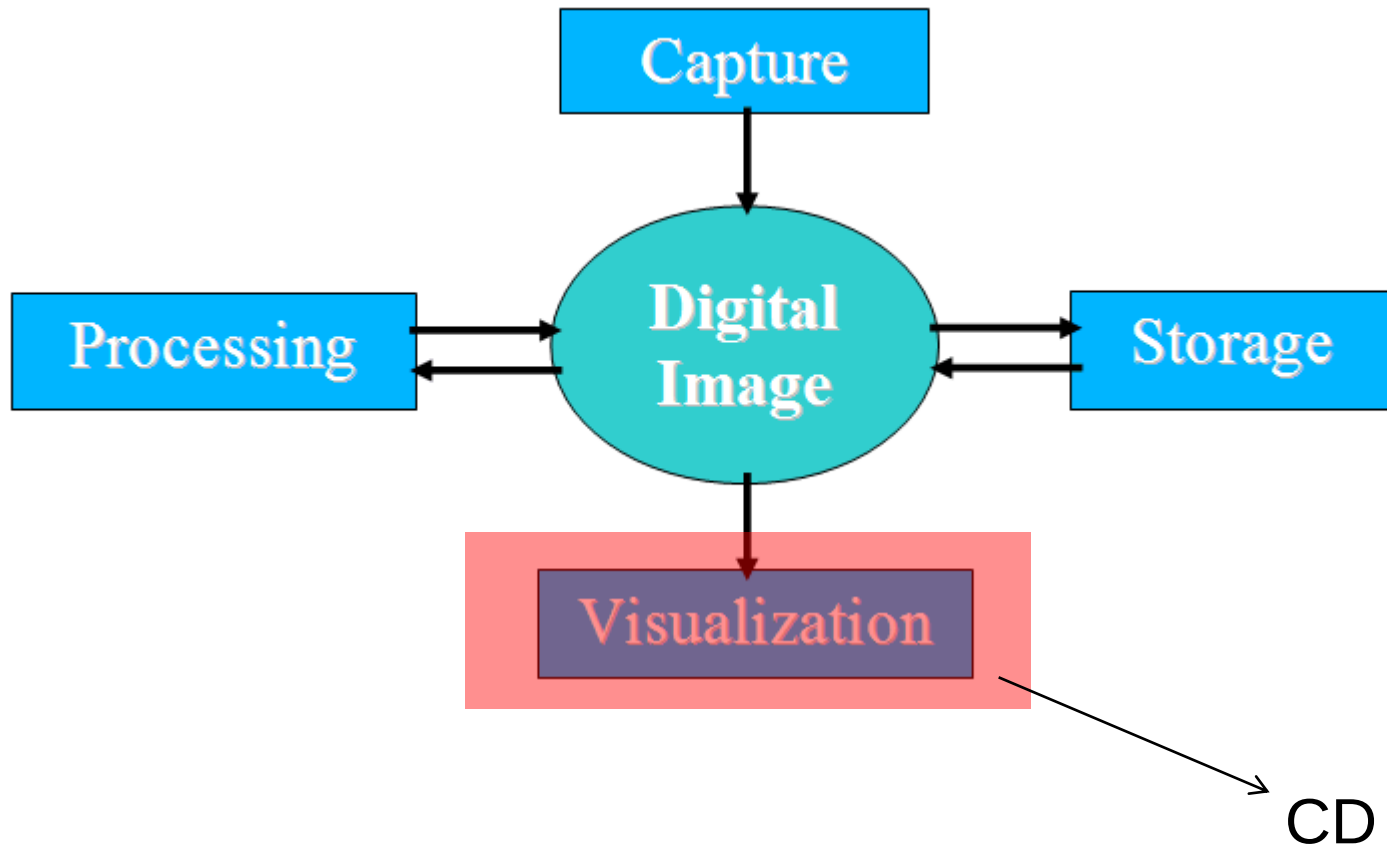
Motif em MWM

GTK em Gnome



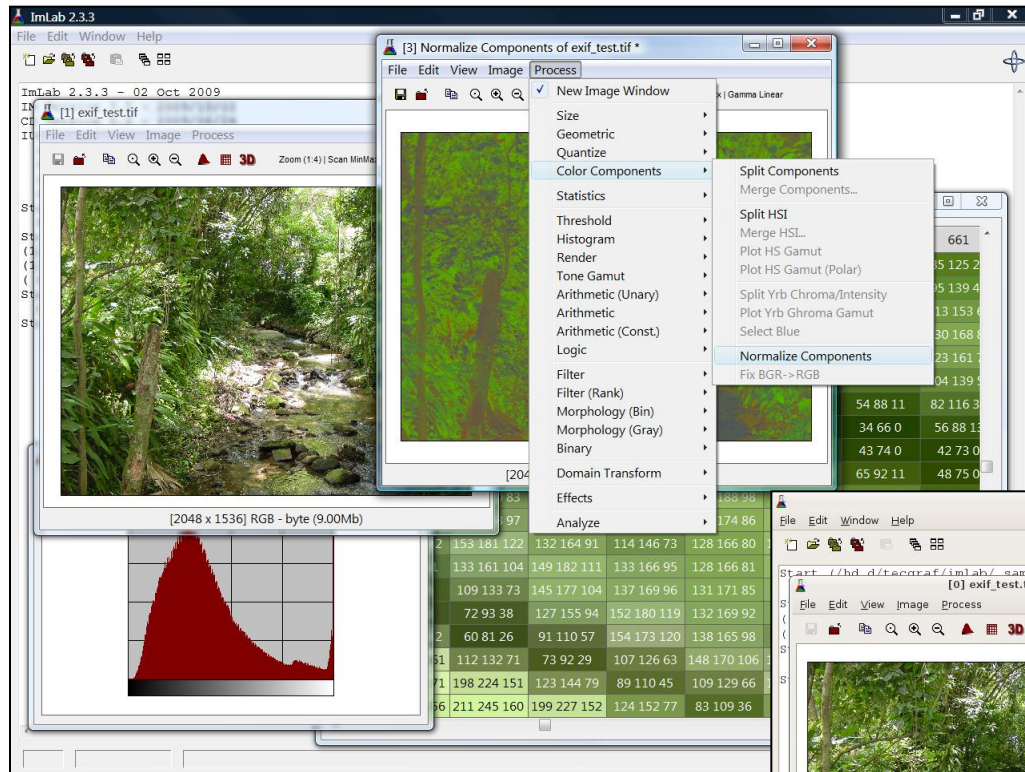
Windows Vista



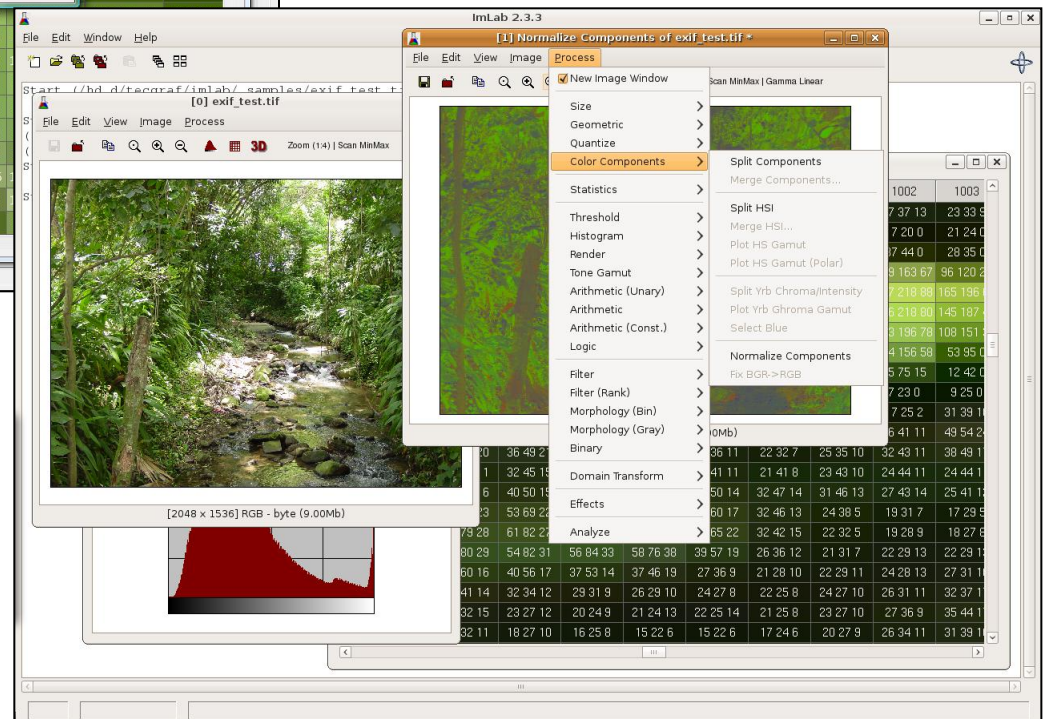


Exemplo de uma Aplicação Completa

Windows Vista



GTK em Gnome





- **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.



```
void main(void)
{
    if (IupOpen() == IUP_ERROR)
    {
        fprintf(stderr, "Error Opening IUP.")
        return;
    }

    ...

    IupMainLoop();
    IupClose();
}
```



- Criação
 - `IupDialog`
 - `IupSetAttribute`
 - `IupSetCallback`
- Criação no Sistema Nativo
 - `IupMap(dlg)` ou
 - `IupShow(dlg)`
- Destruição
 - `IupDestroy(dlg)`

Dialog

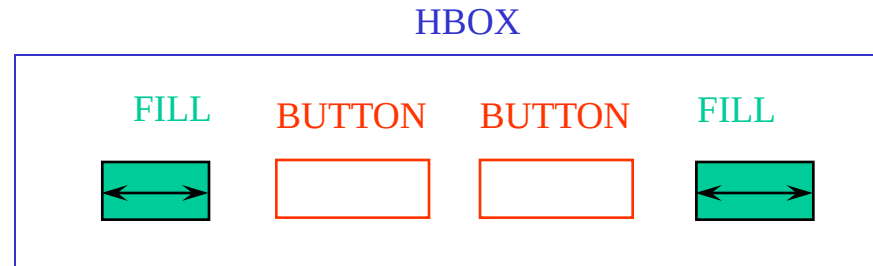
Hbox

Fill

Button

Button

Fill





Concreto

- Mais utilizado (Microsoft, Borland, Delphi, VB).
- Dependente da resolução do sistema.
- Ao variar o tamanho da janela os objetos ficam desposicionados.
- Ferramenta de especificação interativa ("visual") se torna mais simples .

Abstrato

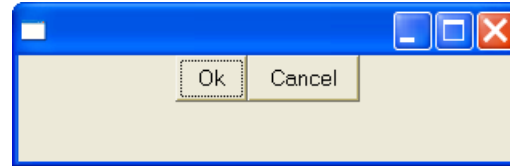
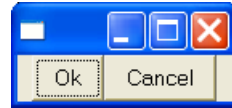
- Menos utilizado.
- Independente da resolução do sistema.
- Rearranjo automático dos elementos após alteração do tamanho da janela.
- Sintaxe mais simples e agradável.
- Manutenção mais simples.
- Ferramenta de especificação interativa ("visual") se torna mais complexa.



```
Ihandle* f1 = IupFill();  
Ihandle* btok = IupButton("Ok", "do_ok");  
Ihandle* btcancel = IupButton("Cancel", "do_cancel");  
Ihandle* f2 = IupFill();  
  
Ihandle* box = IupHbox(f1, btok, btcancel, f2, NULL);  
  
Ihandle* dlg = IupDialog(box);
```



```
Ihandle* dlg = IupDialog
(
    IupHbox
    (
        IupFill(),
        IupButton("Ok", "do_ok"),
        IupButton("Cancel", "do_cancel"),
        IupFill(),
        NULL
    )
);
```

- Usando alguns atributos para melhorar a aparência:

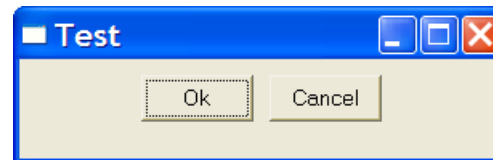
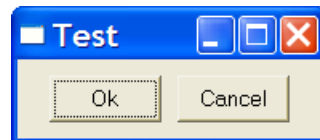
```
IupSetAttribute(box, "MARGIN", "10x10");
```

```
IupSetAttribute(box, "GAP", "10");
```

```
IupSetAttribute(btok, "SIZE", "40x");
```

```
IupSetAttribute(btcancel, "SIZE", "40x");
```

```
IupSetAttribute(dlg, "TITLE", "Test");
```





- TITLE
- VALUE (depende do controle: texto, número, nome, índice, ON/OFF,...)
- SIZE ("widthxheight" ex: "300x300")
 - Escala horizontal é $\frac{1}{4}$ da largura média de um caracter da fonte selecionada.
 - Escala vertical é $\frac{1}{8}$ da altura média de um caracter da fonte selecionada.
 - Também é o tamanho mínimo
- RASTERSIZE ("widthxheight" em pixels)
- EXPAND (YES, NO)
- ACTIVE (YES, NO)
- VISIBLE (YES, NO)
- BGCOLOR ("255 0 128")
- FGCOLOR ("192 192 192")
- FONT ("Times, 12")



```
Ihandle* dialog = IupSetAttributes(IupDialog
(
  IupSetAttributes(IupHbox
  (
    IupFill(),
    IupSetAttributes(IupButton("Ok", "do_ok"), "SIZE=40"),
    IupSetAttributes(IupButton("Cancel", "do_cancel"), "SIZE=40"),
    IupFill(),
    NULL
  ), "MARGIN=15x15, GAP=10")
), "TITLE = Test");
```

Callbacks



```
int btok_action(Ihandle* self)
{
    char color[20];
    strcpy(color, IupGetAttribute(self, "FGCOLOR"));
    IupStoreAttribute(self, "FGCOLOR", IupGetAttribute(self, "BGCOLOR"));
    IupStoreAttribute(self, "BGCOLOR", color);
    return IUP_DEFAULT;
}

int btcancel_action(Ihandle* self)
{
    return IUP_CLOSE;
}

...
IupSetAttribute(btok, "BGCOLOR", "255 0 0");

IupSetFunction("do_ok", btok_action);
IupSetFunction("do_cancel", btcancel_action);
ou
IupSetCallback(btok, "ACTION", btok_action);
IupSetCallback (btcancel, "ACTION", btcancel_action);
```



```
imImage* image = imFileImageLoad(file_name, 0, &error);
```

```
if (err != IM_ERR_NONE)  
    error...
```

```
imImage* new_image = imImageCreateBased(image, x2-x1+1, y2-y1+1, -1, -1);  
imProcessCrop(image, new_image, x1, y1);
```

```
imFileImageSave(file_name, "JPEG", new_image);  
imImageDestroy(new_image);  
imImageDestroy(image);
```



```
cdCanvas* canvas = cdCreateCanvas(CD_PS, "cdtest.ps");

cdCanvasMarkSize(canvas, 40);
cdCanvasMark(canvas, x, y);

cdCanvasFont(canvas, "Courier", CD_PLAIN, 12);
cdCanvasTextAlignment(canvas, CD_CENTER);
cdCanvasText(canvas, x, y, text);

cdCanvasGetTextBox(canvas, x, y, text, &xmin, &xmax, &ymin, &ymax);
cdCanvasRect(canvas, xmin, xmax, ymin, ymax);

cdCanvasKill(canvas);
```

Antonio Scuri
scuri@tecgraf.puc-rio.br

