

# Programação Orientada a Eventos em Sistemas de Interfaces com o Usuário

**Luiz Fernando Martha**

**Departamento de Engenharia Civil – PUC-Rio**

**CIV2802 – Sistemas Gráficos para Engenharia**

(adaptado de Marcelo Gattass)

# Processo de Interação (Eng. Cognitiva)

---

## Golfo de Execução:

- formulação da intenção;
- especificação da ação;
- execução.



## Golfo de Avaliação:

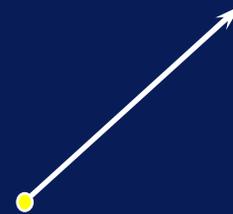
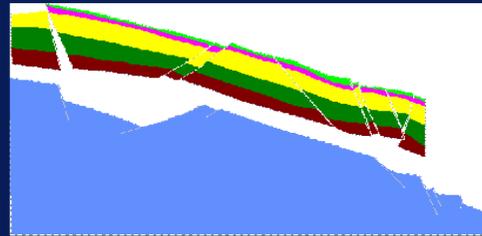
- percepção;
- interpretação;
- avaliação.

Norman, 1986

# Signos (Semiótica)

---

■ *Índices*



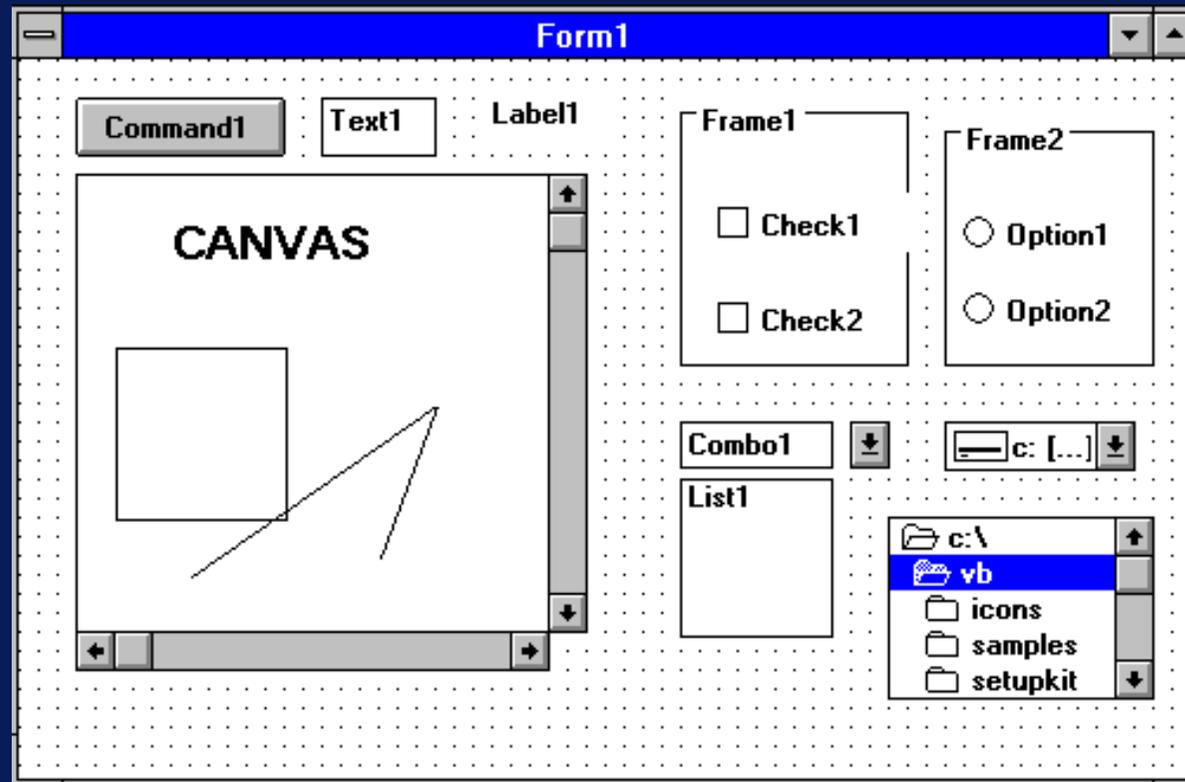
■ *Ícones*



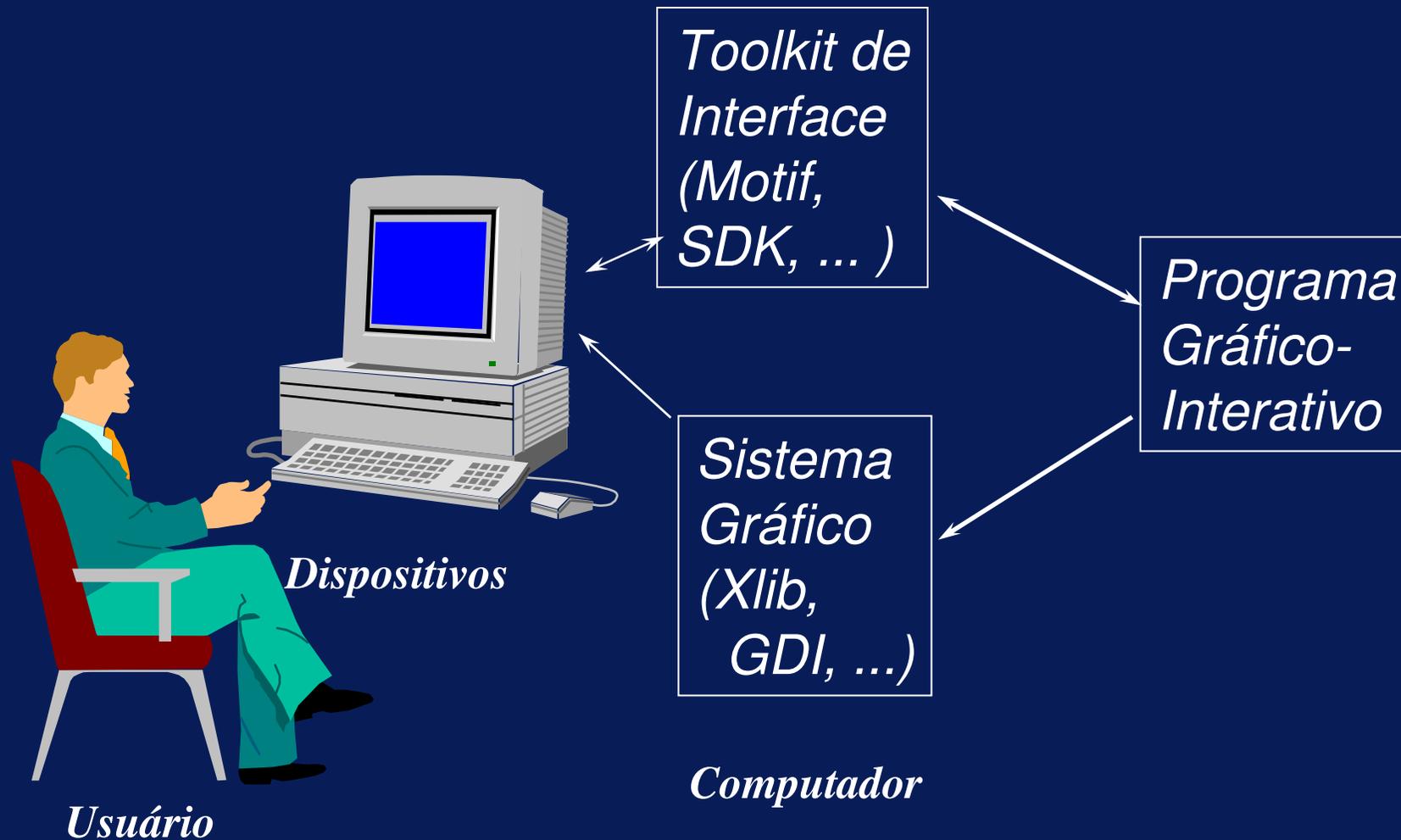
■ *Símbolos*

A

# Objetos de comuns interface



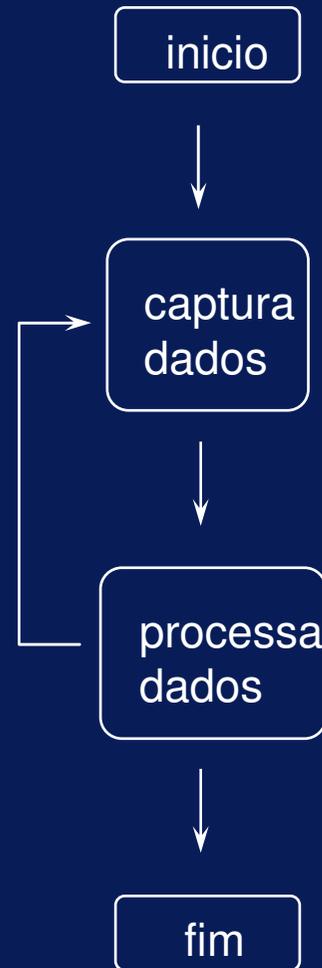
# Modelo de Programação



# Programação Convencional

---

**Programação Convencional**  
Os comandos são executados segundo uma ordem pré-estabelecida e seqüencial.

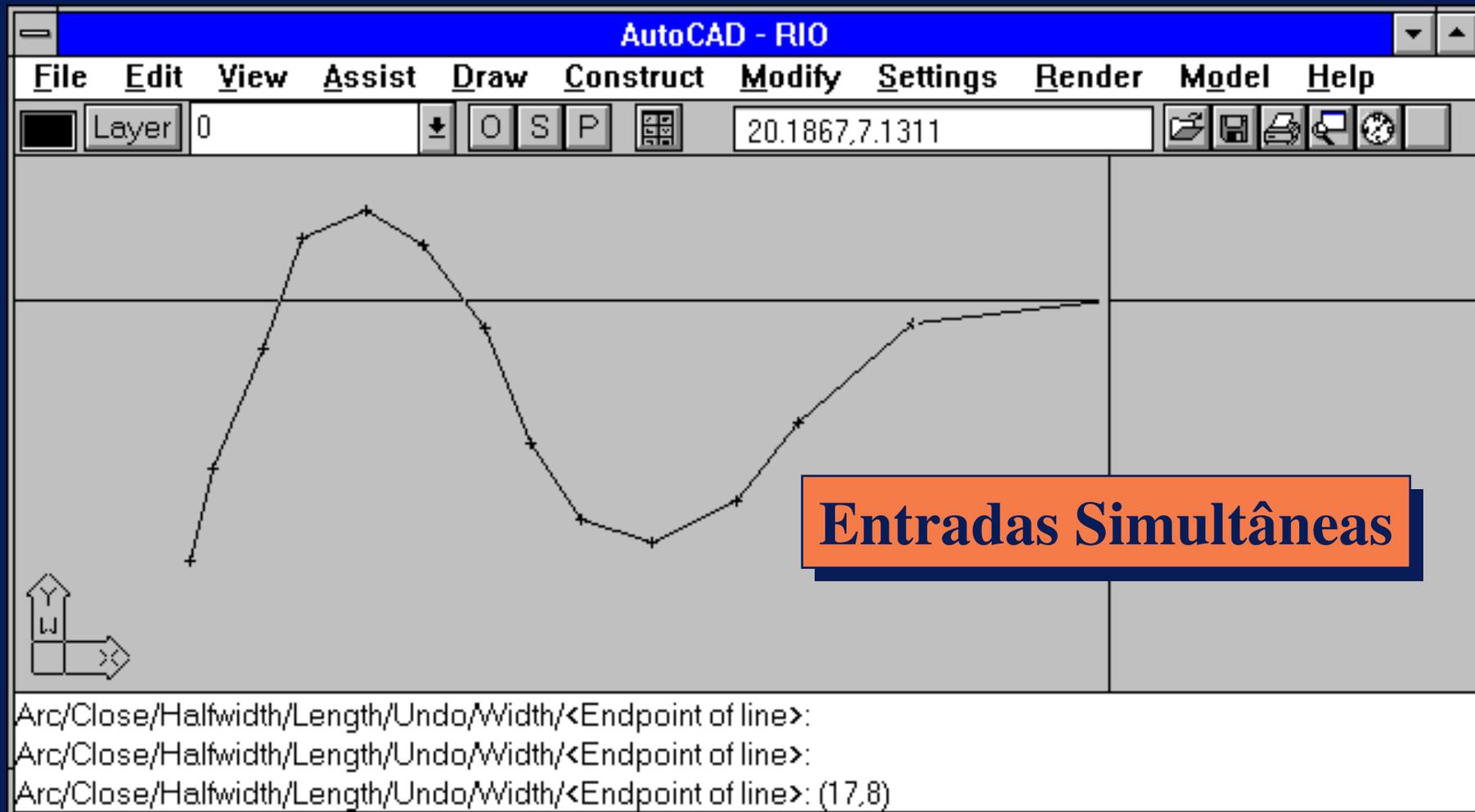


# Técnicas de Interação

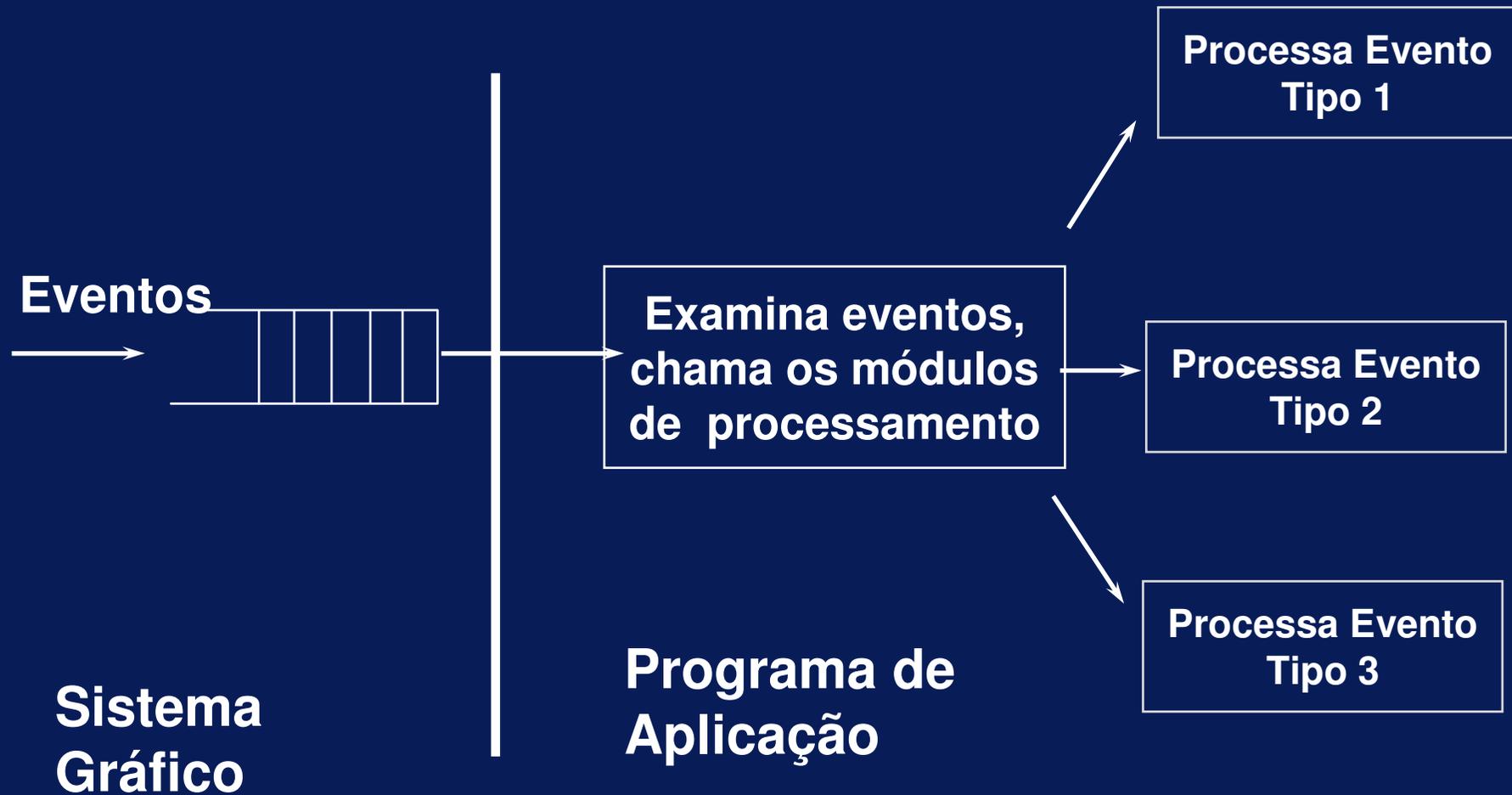
---

- Solicitação (*Request*)
- Amostragem (*Sample*)
- Eventos (*Event*)
  - eventos [Xlib, SDK]
  - *callbacks* [Motif, IUP, Visual...]
  - *listeners* [Java/OO]
  - *signals & slots* [Qt]

# Problemas com solicitação



# Modelo de Eventos



# Eventos típicos (WIMP)

---

**KeyPress**

**KeyRelease**

**ButtonPress**

**ButtonRelease**

**Motion**

**LeaveNotify**

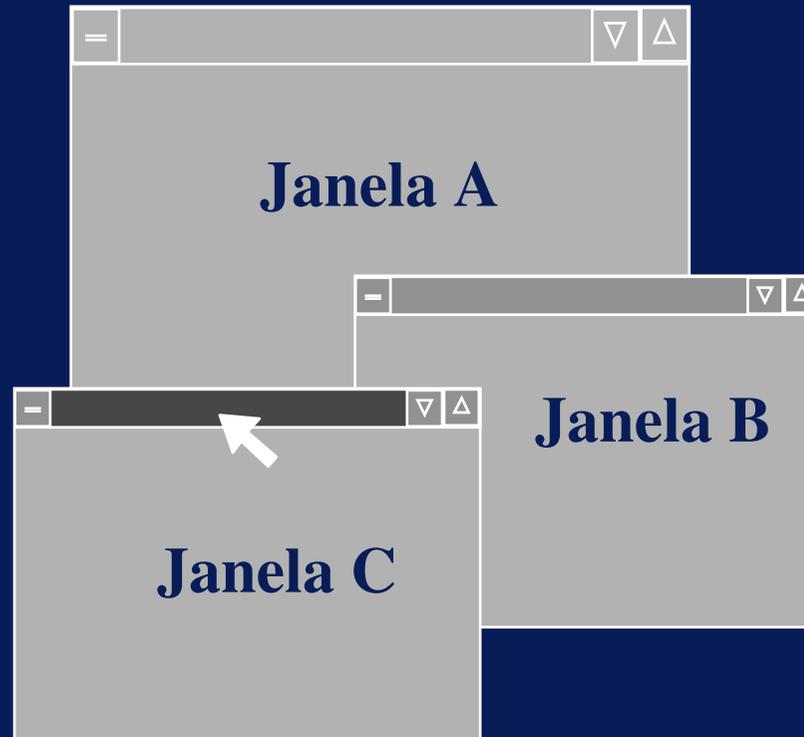
**EnterNotify**

**WindowExposure**

**Resize**

**Timer**

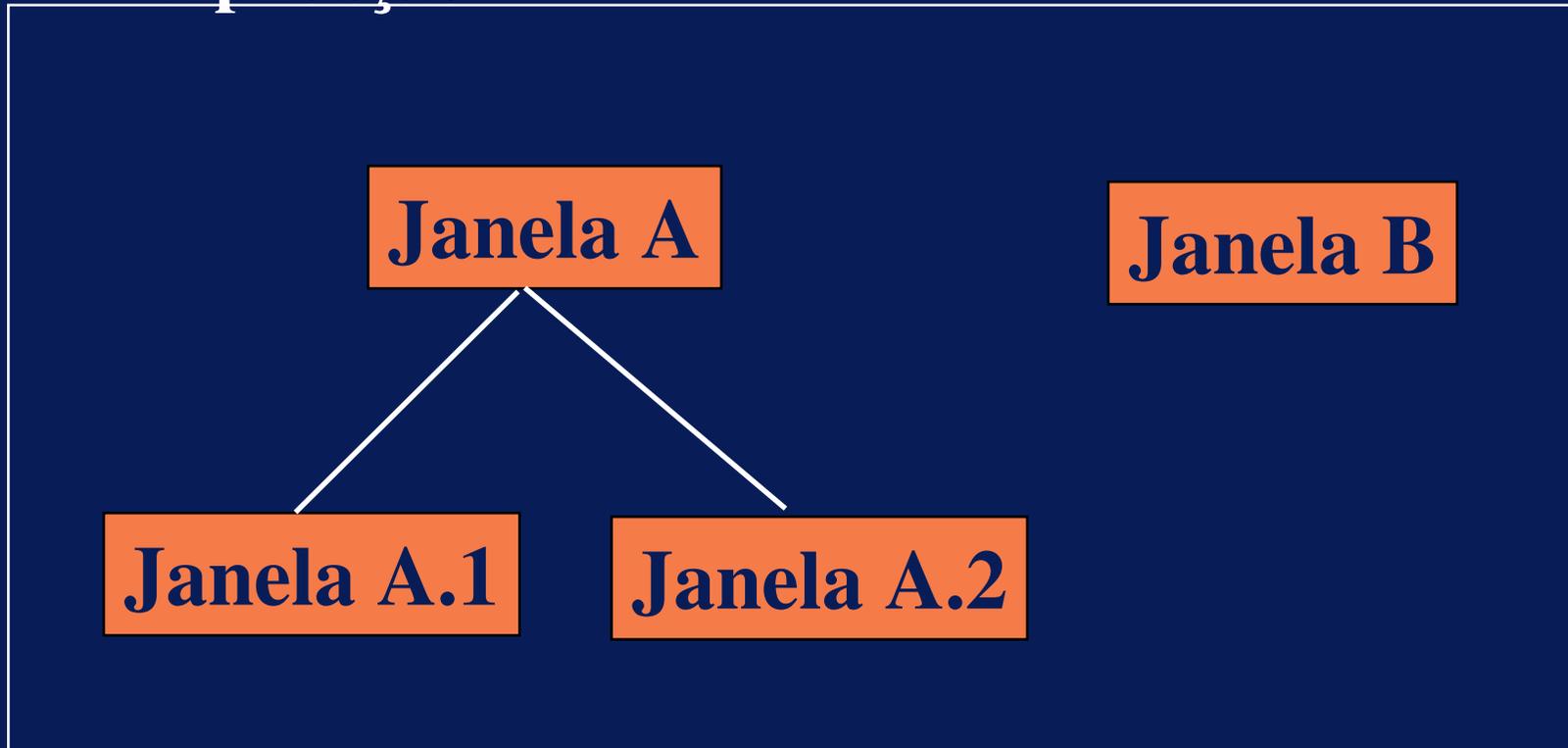
**Idle**



# Janelas e tratadores de eventos

---

Aplicação

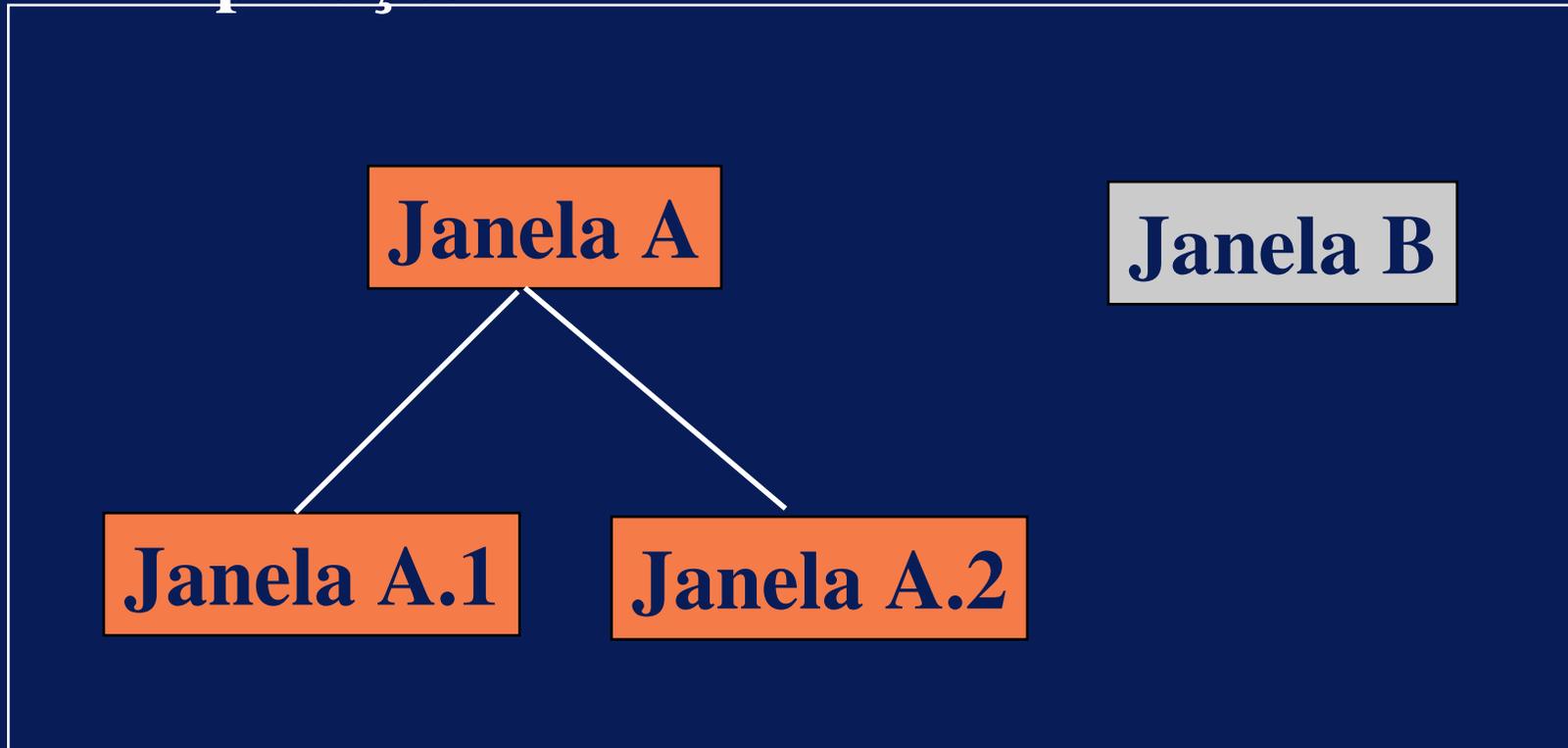


■ Geralmente um tratador de eventos por aplicação

# Janelas e tratadores de eventos

---

Aplicação

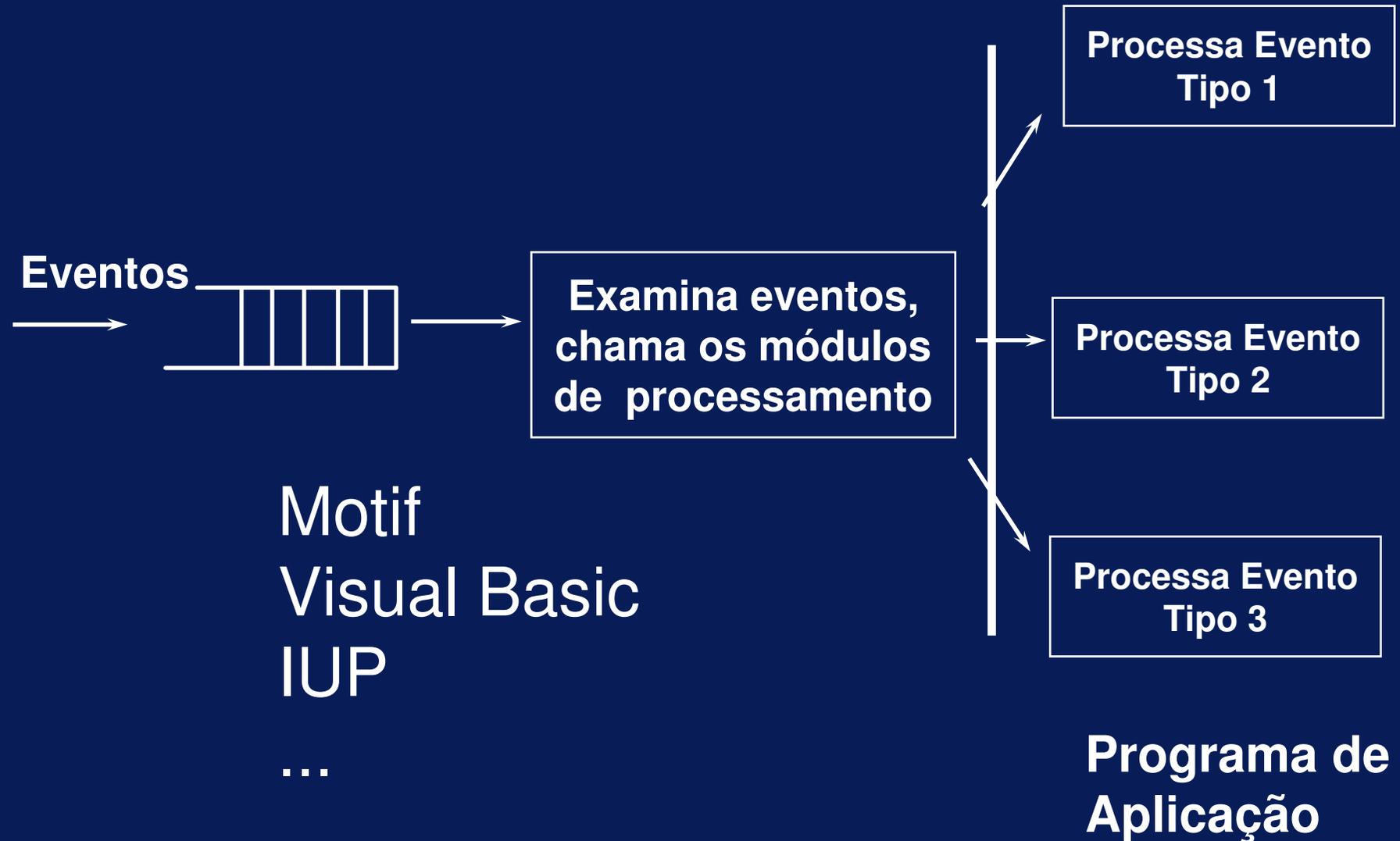


Tratador de eventos I



Tratador de eventos II

# Modelo de *Callbacks*



```
static int repaint (Ihandle *self)
{ ... }
```

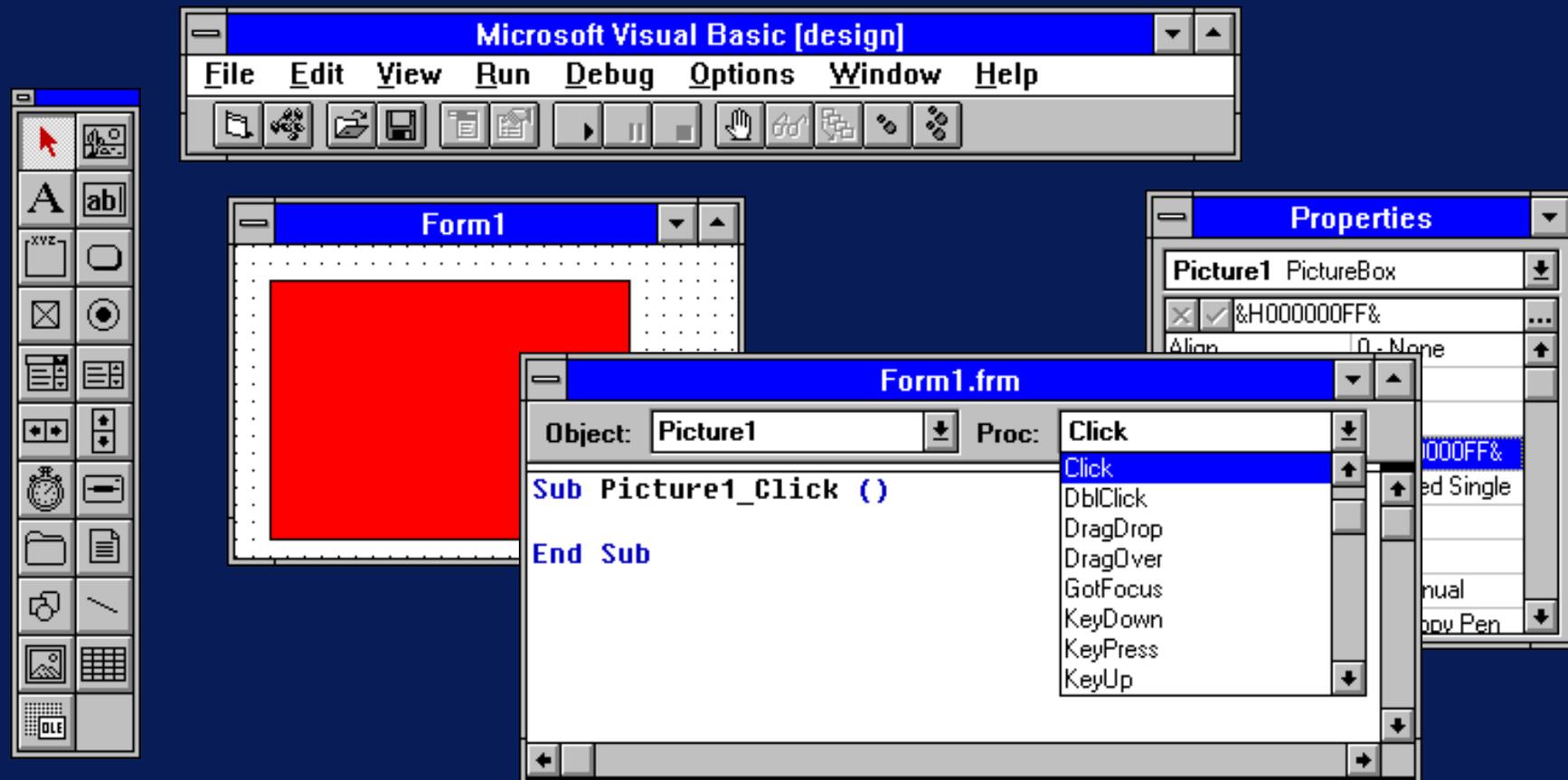
IUP/LED

```
void main (void)
{
  IupOpen( );
  IupLoad ("canvas.led");
  IupSetFunction ("a_repaint", (Icallback)repaint);
  IupShow (IupGetHandle("main"));
  IupMainLoop( );
  IupClose( );
}
```

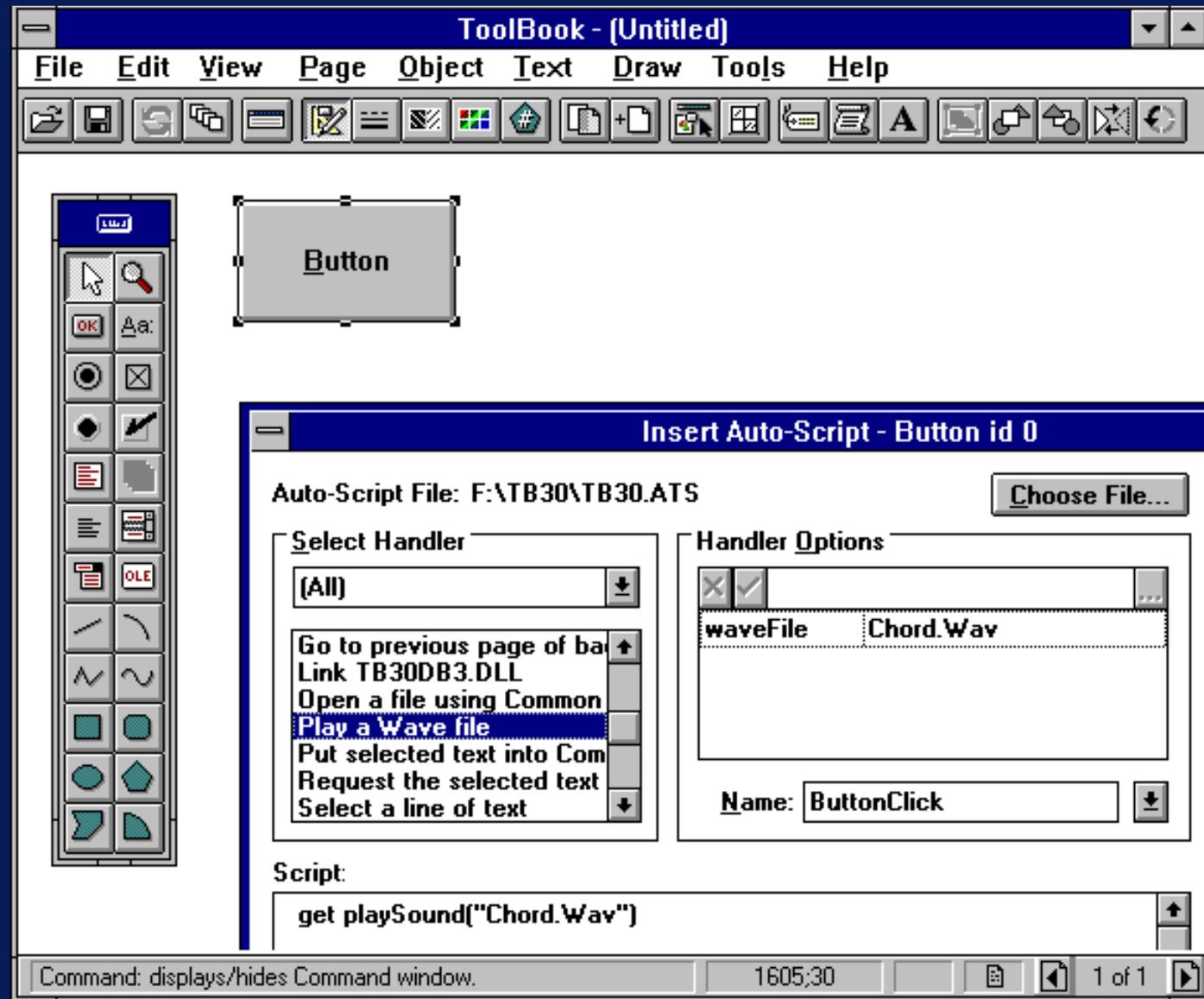
canvas.led

```
main = dialog [TITLE="IUP Canvas"] ( canvas (a_repaint) )
```

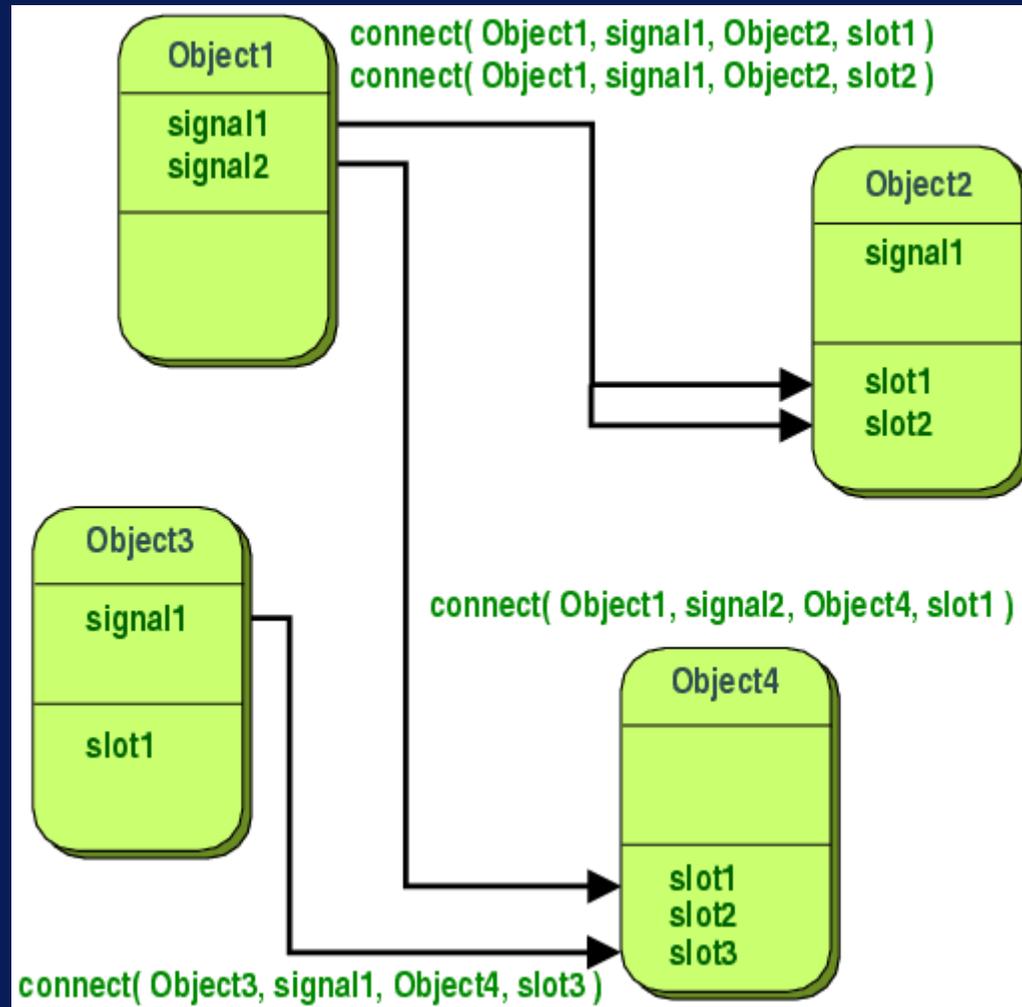
# Visual Basic



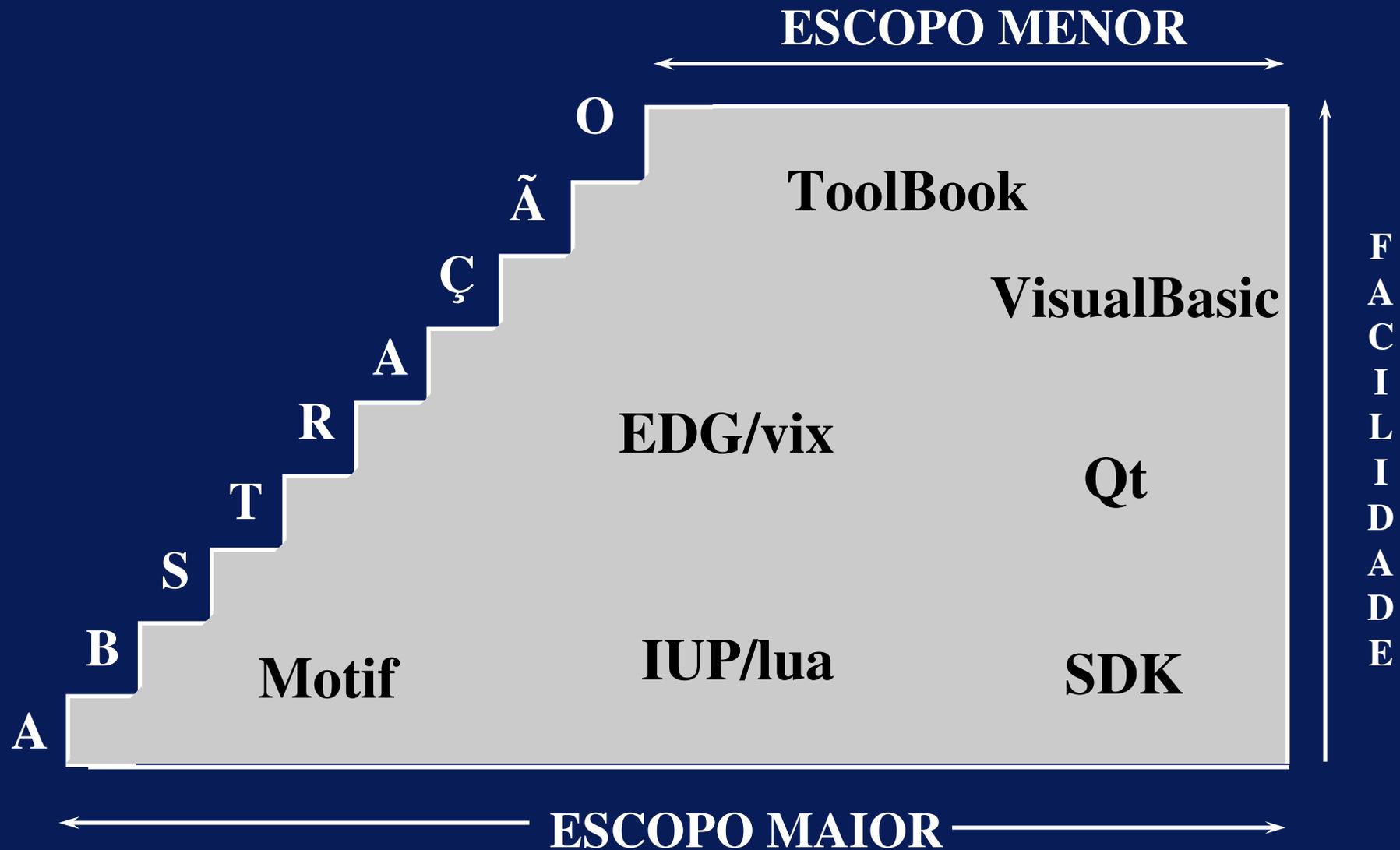
# ToolBook



# Modelo de *Signals & Slots* [Qt]



# Nível de abstração e escopo



**Em que usar o que?**

# Modelo do IUP/LED

---

- Aplicação = conjunto de diálogos
- Diálogos = hierarquia de elementos de interface
- Especificação de *layout*
  - Concreto X Abstrato
- Atributos definem a aparência

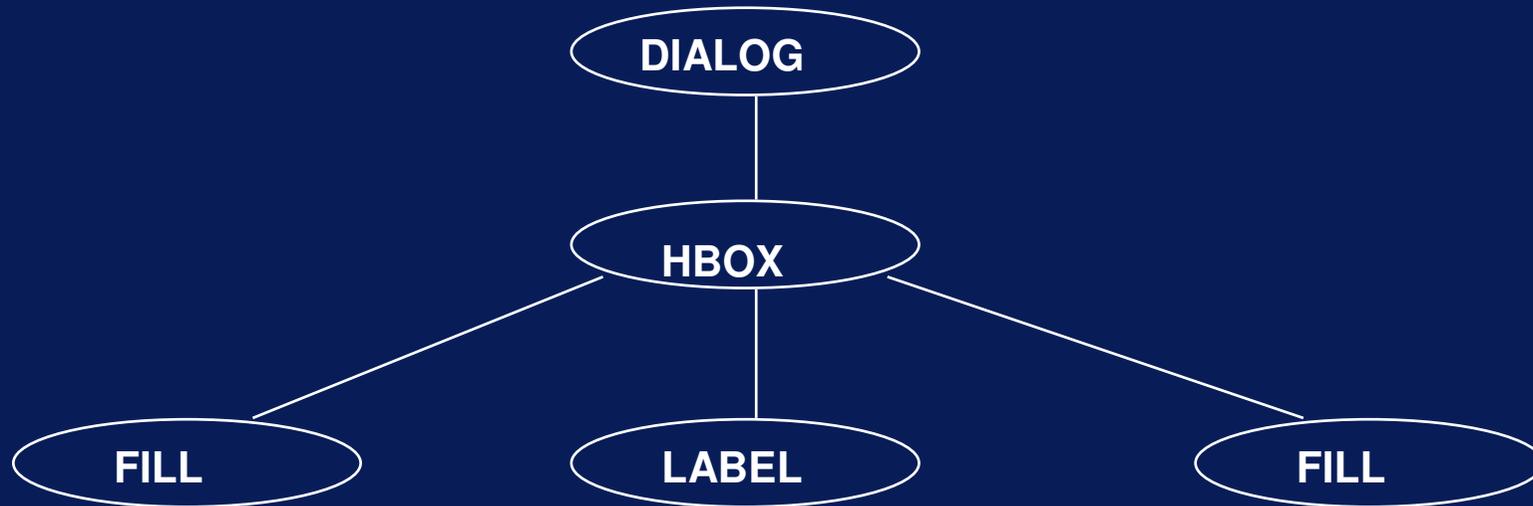
# Elementos de Interface

---

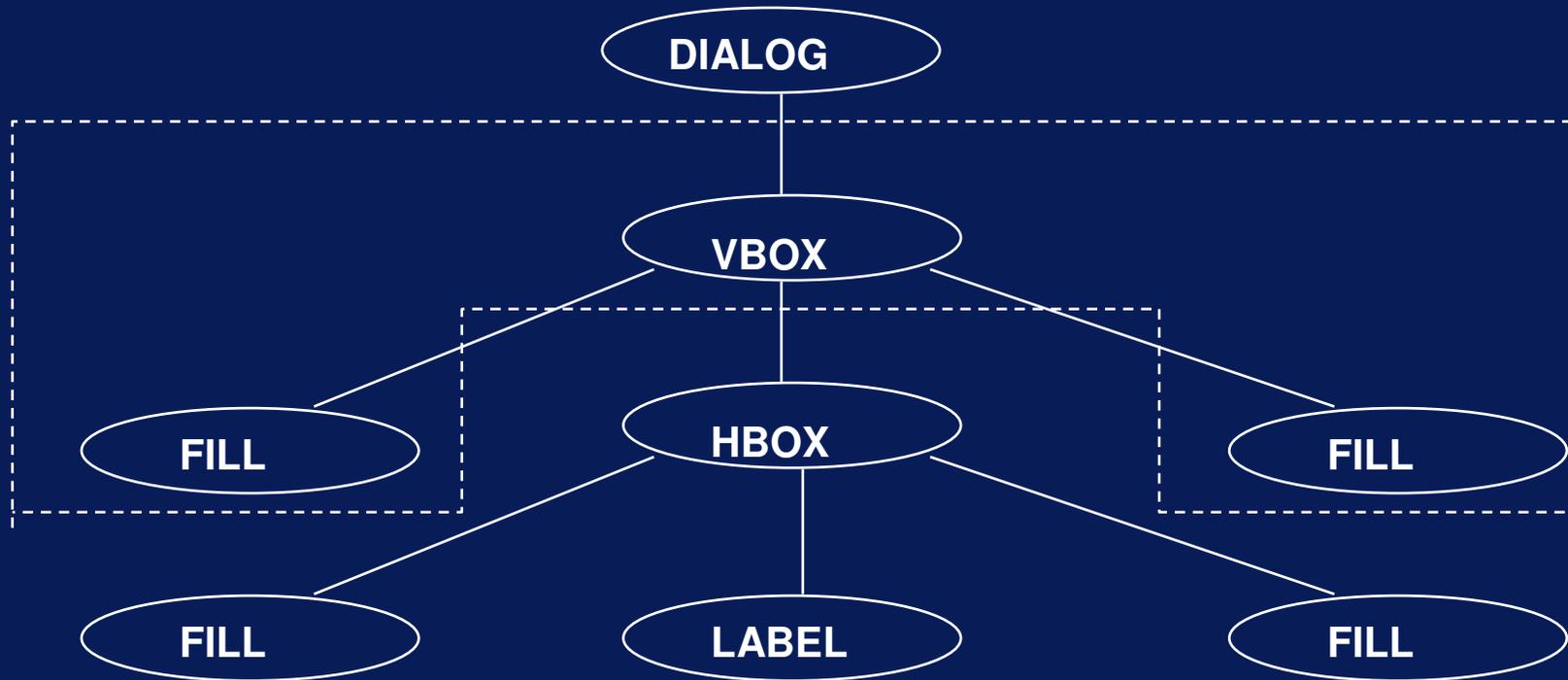
- **Primitivos**
  - Button, Canvas, Frame, Image, Item, Label, List, Submenu, Text, Toggle, Matrix, Multiline
- **Agrupamento**
  - Dialog, Radio, Menu
- **Composição**
  - Hbox, Vbox, Zbox
- **Preenchimento**
  - Fill

# Composição do *layout*

---



# Centralizando elementos



Teste

Texto Centralizado

# Layout abstrato

```
the_menu = ...
```

```
the_canvas = CANVAS[ BUTTON_CB = button_cb,  
                     MOTION_CB = motion_cb](repaint_cb)
```

```
the_dialog = DIALOG[ MENU=the_menu ]
```

```
(  
  VBOX(  
    FILL(),  
    HBOX(  
      FILL(),  
      the_canvas,  
      FILL()  
    ),  
    FILL()  
  )  
)
```

