

Interfaces con el Usuario

David Cabrero Souto

Facultad de Informática
Universidade da Coruña

Curso 2009/2010



Lectura recomendada:

- <http://arstechnica.com/articles/paedia/gui.ars>



Lectura recomendada:

- <http://arstechnica.com/articles/paedia/gui.ars>

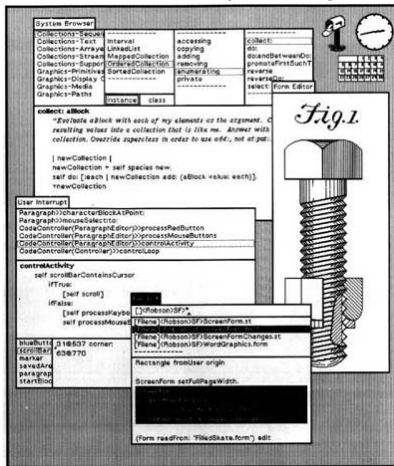


Demo de 1968



Lectura recomendada:

- <http://arstechnica.com/articles/paedia/gui.ars>



- Apple. Fundada en 1976 por Steve Jobs.
 - Apple][. Primer sistema.
 - Mac OS 9. IU integrada en el sistema operativo.
 - Mac OS X. Sabor a BSD.
- Win32. Hijo putativo de MacOS.
- Unix. X-Window.



- Cliente/Servidor.
- Servidor.
 - Networking.
 - Drivers: tarjeta vídeo, input devices, ...
 - Varios displays (`maquina:0.0`)
 - Ventana = rectángulo
- Clientes.
- Gestor de ventanas / Entorno de escritorio.
 - Ventana = rectángulo + decoraciones. Se puede mover, cambiar de tamaño, iconificar, ...
- Toolkit gráfico.



Toolkit gráfico

**Gestor de
ventanas**

Xlib

Servidor X



- TV
 - PVR. mythtv
 - STB. amino, mhp
 - Consola videojuegos
- PDA, teléfono móvil



¿ Lo siguiente en X-Window ?

- Servidor que renderiza sobre GL
- Gestor de ventanas que aprovecha las posibilidades de GL
- Demo de XLG y compiz



- **Librería de widgets y objetos relacionados**

- Widget = window + gadget

- Algunas librerías:

- Swing, SWT (JAVA)
- GTK [GNOME] (C) (Bindings para perl, python, ruby, c++, caml, erlang, ...)
- QT [KDE] (C++)



- **Librería de widgets y objetos relacionados**
 - **Widget = window + gadget**
- Algunas librerías:
 - Swing, SWT (JAVA)
 - GTK [GNOME] (C) (Bindings para perl, python, ruby, c++, caml, erlang, ...)
 - QT [KDE] (C++)



- Librería de widgets y objetos relacionados
 - Widget = window + gadget
- Algunas librerías:
 - Swing, SWT (JAVA)
 - GTK [GNOME] (C) (Bindings para perl, python, ruby, c++, caml, erlang, ...)
 - QT [KDE] (C++)



- 1 Inicializar el objeto que representa la ventana, aplicación, top-level, . . .
- 2 Añadir widgets
- 3 Añadir comportamiento (reaccionar a eventos)
- 4 Mostrar la ventana
- 5 Bucle de eventos



HelloWorld: SWING

```
import javax.swing.JFrame;
import javax.swing.JLabel;

public class HelloWorldSwing {
    public static void main(String[] args) {
        JFrame frame = new JFrame("HelloWorldSwing");
        final JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
```



```
#include <gtk/gtk.h>

int main( int   argc,
          char *argv[] )
{
    GtkWidget *window;

    gtk_init (&argc, &argv);

    window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
    gtk_widget_show (window);

    gtk_main ();

    return 0;
}
```



```
#include <qapplication.h>
#include <qpushbutton.h>

int main( int argc, char **argv )
{
    QApplication a( argc, argv );

    QPushButton hello( "Hello world!", 0 );
    hello.resize( 100, 30 );

    a.setMainWidget( &hello );
    hello.show();
    return a.exec();
}
```




```
import org.eclipse.swt.widgets.*;

public class HelloWorld {
    public static void main(String arg[]) {
        Display display = new Display();
        Shell shell = new Shell(display);
        shell.setText("Hello World");
        shell.open();
        while(!shell.isDisposed()) {
            if (!display.readAndDispatch()) {
                display.sleep();
            }
        }
        display.dispose();
    }
}
```



```
#!/usr/bin/env python

import gtk

window = gtk.Window(gtk.WINDOW_TOPLEVEL)
window.set_title("Hola mundo")
window.show()

gtk.main()
```



- Programación dirigida por eventos
 - El programador no establece el flujo del programa
 - Los sucesos generan eventos
 - Controlador/Manejador de eventos
 - Muy importante el concepto de estado
- Patrón MVC (Model-View-Controller)



- Programación dirigida por eventos
 - El programador no establece el flujo del programa
 - Los sucesos generan eventos
 - Controlador/Manejador de eventos
 - Muy importante el concepto de estado
- Patrón MVC (Model-View-Controller)



Comportamiento: Contraejemplo

```
class Stack
{
    private float[] elements = new float[100];
    private int topOfStack = -1;

    public void push() {
        float item =
            Console.readFloat("Type a float to push: ");
        elements[++topOfStack] = item;
    }

    public void pop() {
        Console.println("Top of stack item: " +
            elements[topOfStack--]);
    }
}
```



- Sistema de callbacks.
- Se asigna una función/método a los evento relevantes.

```
def hola(widget, data=None)
    print "¡Hola Mundo!"
...
button = gtk.Button("Hola mundo")
button.connect("clicked", hola, None)
...
```



- No nos interesa posicionamiento absoluto
- Compuestos, contenedores, paneles, ...
 - Elementos visibles/invisibles
 - Estructura en árbol
 - Determinan la disposición de sus hijos

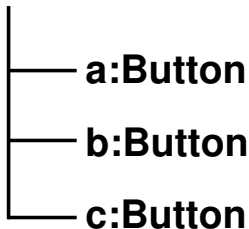


a:Button

b:Button

c:Button

ROW



- **Contenedor:** *caja* (box)
- **Clases:** `gtk.HBox()`, `gtk.VBox()`
- **Métodos:** `pack_start()` y `pack_end()`

Ejemplo

```
...
box = gtk.HBox()
buttonA = gtk.Button("botón A")
box.pack_start(buttonA)
buttonB = gtk.Button("botón B")
box.pack_start(buttonB)
buttonA.show()
buttonB.show()
box.show()
window.add(box)
...
```

- **Contenedor:** *tabla*
- **Clases:** `gtk.Table()`
- **Métodos:** `attach()`

Ejemplo

```
...  
table = gtk.Table(2,2)  
buttonC = gtk.Button("botón 1")  
table.attach(buttonC,0,1,0,1)  
buttonD = gtk.Button("botón 2")  
table.attach(buttonD,1,2,0,1)  
buttonE = gtk.Button("botón 3")  
table.attach(buttonE,0,2,1,2)  
buttonC.show()  
buttonD.show()  
buttonE.show()  
table.show()
```



- Concepto de foco: widget seleccionado.
- Los eventos de teclado se envían al widget que tiene el foco.
- No confundir con el puntero del ratón.
- Cambiar el foco:
 - Usando el ratón.
 - Usando el teclado: *mod*+TAB, cursores.
- *Tab order*.
- Algunos widgets tienen cursor de texto.
- *Foco de ventana*.



- En una librería OO, cada widget tiene métodos y/o atributos para recuperar la información introducida por el usuario.
- Tb. de forma implícita en los eventos.

Python

```
...  
entry = gtk.Entry()  
...  
entry.get_text()  
...
```

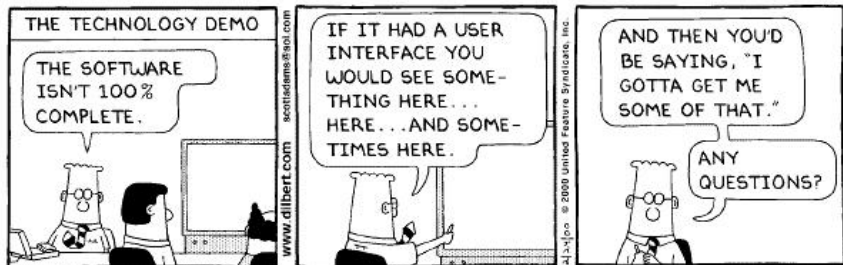
SWT

```
...  
Text text = new Text(parent, SWT.SINGLE | SWT.BORDER);  
...  
text.getText()  
...
```

- “*Object-Oriented Interface Design. IBM Common User Access Guidelines*”. 1992.
- FreeDesktop
http://www.freedesktop.org/wiki/Standards_2ficon_2dnaming_2dspec
http://tango.freedesktop.org/Tango_Desktop_Project



Qué NO hacer



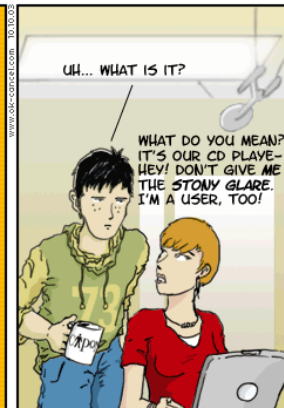
Copyright © 2000 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited



Qué NO asumir



OK/Cancel



Kicking the Llama : copyright 2003 tom chi and kevin cheng