# Introduccion a la Ingenieria de Interfaces Graficos de Usuario basada en TCL y TK

Santiago Pavón

# Introduccion

- Desarrollo de Interfaces X.
- Problematica.
- Soluciones
- Solucion Tcl-Tk.

# Desarrollo de Interfaces

- Aplicaciones basadas en ventanas son dificiles de programar debido a la complejidad añadida por la parte grafica..
- Elevada complejidad de las librerias.
- Caracteristicas perseguidas:
  - Rapido desarrollo de las aplicaciones.
  - Adaptables gustos de los usuarios.
  - Extensibles.
  - etc...
- X permite configurar parametros para adaptar a gustor particulares, permite extender el conjunto de widget,...
- Soluciones tiempo de desarrollo:
  - Generadores automaticos de aplicaciones.
  - Lenguajes programacion de alto nivel.
  - Especificaciones de interfaces.

# Ejemplo: Dirt

- Editor interactivo de interfaces graficas de usuario.
- Basado en Xt.
- Permite construir el interface de la aplicacion paso a paso, visualizando el resultado final inmediatamente.
- Establecer asociaciones con las funciones C a ejecutar.
- Salida es un programa en C que debe encadenarse con las funciones escritas por el usuario.

# Ejemplo:Wcl

- Widget Creation Library.
- Especificacion del interface de usuario en un fichero de recursos.
- Esta especificacion se interpreta en tiempo de ejecucion.
- La aplicacion se contruye encadenendo la libreria `Wcl` con las funciones de usuario.

# Ejemplo:Wcl (cont)

```
*simul.title:  SIMUL

*simul*Label.BorderWidth:   0
*simul*Toggle*ShapeStyle:   roundedRectangle
*simul*Toggle.Translations: #override \
    <Btn1Down>,<Btn1Up> :   set()notify()


*simul.wcChildren: simulpand
*simul.wcManaged:  false
*simul.borderWidth: 0

*simulpand.wcClass:    panedWidgetClass
*simulpand.wcChildren: simulbar,      \
                       Parameters,    \
                       simulcommands, \
                       SimulCommandOut

...

*simulTOPO.wcClass:    commandWidgetClass
*simulTOPO.label:      TOPO
*simulTOPO.callback:   TOPOCB
```

# Ejemplo: XF

- Es un entorno para desarrollar interfaces graficos de usuario.
- Permite construir o modificar las aplicaciones mientras estan ejecutandose.
- Tambien permite definir las respuestas de la aplicacion ante los eventos ocurridos o las acciones del usuario.
- Basado en Tcl-Tk.
- Interprete.

# Que es Tcl-Tk

- Tcl: lenguaje comandos utilizable aplicaciones C.
- Tk: Toollkit X11 con widget basados en Tcl.

Proporciona un lenguaje interpretado que permite controlar todos los aspectos de una aplicacion interactiva.

Eleva y simplifica el nivel de programacion en X.

Aumenta la velocidad de desarrollo.

Gran potencia:

- mas cosas son programables.
- colaboracion entre aplicaciones.

# Tcl: Tool Command Language

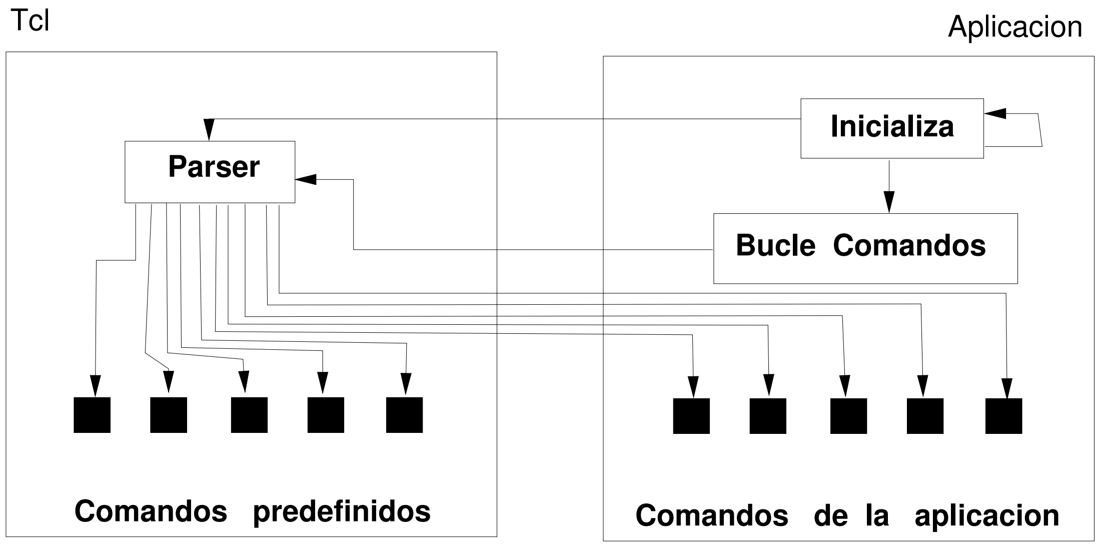
Problema:

- Aplicaciones interactivas necesitan lenguajes de comandos.
- Normalmente cada aplicacion se crea el suyo.
- Resultado: debiles, peculiares.
- Emacs y tcsh tienen buenos interpretes de comandos, pero no se puede reutilizar.

Solucion: Tcl.

- lenguaje de comandos es una libreria C.
- Caracteristicas: procedures, variables, listas, expresiones, bucles, etc.
- Extensible por las aplicaciones.

# Integracion de Tcl en Apl. C




- La aplicacion genera scripts Tcl.

- Tcl analiza (parser) scripts y llama a los procedimientos asociados a cada comando.

- La aplicacion extiende los comandos predefinidos de Tcl:
  - Define nuevos tipos de objetos en C.
  - Implementa las primitivas sobre los objetos como nuevos comandos Tcl.
  - Crea comandos complejos como scripts Tcl.

# Toolkit Tk

Problema:

- Elevada dificultad de contruccion de aplicaciones con buenos interfaces de usuario.

Solucion incorrecta:

- C++, toolkits orientados a objetos.
- Mejora insignificante productividad (10-20%). Se sigue programando a bajo nivel.

Solucion correcta:

- Elevar el nivel de programacion.
- Crear interfaces escribiendo scripts Tcl.

# Creacion de una interface con Tk

Los widget/ventanas tienen un nombre:

```
.control.fin
```

Los widgets se crean con el comando *nombre de la clase*:

```
button .control.fin -text FIN -command exit
```

Ademas se ha creado el comando `.control.fin`

Decirle al gestor de la geometria donde situar el widget:

```
place .control.fin -x 10 -y 25
pack  .control.fin -side right
```

# Otras caracteristicas de Tk

Manipulacion de los widgets:

```
.control.fin flash
.control.fin configure -foreground red
```

Usar Tcl para interconexiones:

- Un boton llama a un script tcl.
- Una scrollbar y una listbox se comunican mediante Tcl.
- Definir gestion de eventos en Tcl (`bind`).
- Selecciones X accesible via Tcl.

Tk proporciona tambien interfaces para:

- Crear nuevas clases de widgets.
- Crear nuevos gestores de geometria.

# Elementos de una aplicacion basada en Tk

1. El interprete Tcl.

2. El toolkit Tk.

3. Codigo en C especifico de la aplicacion (primitivas tcl):
   - nuevos tipos de objetos.
   - nuevos widgets.

4. Scripts Tcl (composicion de primitivas):
   - Construir el interfaz de usuario.
   - Respuesta a los eventos.

# La aplicacion Tk mas sencilla: WISH

Codigo en C: leer una linea de `stdin` y evaluarla.

Pueden escribirse aplicaciones como scripts de `wish`.

- Hello, world:

```
label .hello -text "Hello, world"
pack  .hello
```

- Navegador:

```
listbox .list -yscroll ".scroll set"
pack    .list -side left

scrollbar .scroll -command ".list yview"
pack      .scroll -side right -fill y

bind .list <Key-q> {puts adios ; destroy .}
bind .list <Double-1> {mover [selection get]}

focus .list

proc llenar {} {
   global dir
   .list delete 0 end
   foreach i [exec ls -aF $dir] {
      .list insert end $i
   }
}

proc mover {marca} {
   global dir
   if {[file isdirectory $dir/$marca]} {
      set dir $dir/$marca
      llenar
   }
}

set dir "."
llenar
```

# Composicion de Aplicaciones

Problema:

- La comunicacion entre aplicaciones es via seleccion X.
- Resultado: aplicacines monoliticas.

Solucion: `send`

- `set apl_nombre comando`
- Una aplicacion puede invocar comandos en otra aplicacion.
- Resultado: comunicacion potente.
- Otras soluciones: Tcl-DP.

# Composicion de Aplicaciones (cont)

Ejemplos:

- Depurador envia comandos al editor para resaltar la linea de ejecucion.
- Panel de control invoca y utiliza varias aplicaciones.

Resultados:

- Construir sistemas complejos a partir hiperherramientas especializadas y reusables.
- Facil creacon de objetos activos (comandos Tcl enpotrados). Ejemplo: hipertexto, hipermedia,...

# Inconvenientes tcl-tk

- Hay que aprender un nuevo lenguaje.

- Limites en las prestaciones de velocidad debido a la interpretacion.

  Pero resultados soprendentes.

- Incompatibilidad interface C con Xt, libreria Motif.

- Tcl no analiza la semantica estatica.

-

# Conclusiones: Interfaces con tcl-tk

- Programacion en alto nivel:
  - menos que aprender.
  - mayor velocidad de desarrollo.

- Un lenguaje interpretado controla todo:
  - las aplicaciones pueden entenderse y modificarse en tiempo de ejecucion.

- 