

Tutorial de Instalación de Kylix3

Por: Pablo Alejandro Zúñiga Muñoz (pabloz@unicauca.edu.co)

Versión de este documento: 1.0

Fecha última revisión: Marzo 15 de 2.004

Kylix es un entorno de programación visual (RAD) compatible con Borland Delphi t Borland C++ Builder. Está Orientado a objetos y utiliza los lenguajes de programación Object Pascal y C++. Es un entorno integrado (IDE), con lo que las tareas de edición, depuración y ejecución se realizan con la misma herramienta.

Kylix cuenta con herramientas propias para la creación de aplicaciones para el acceso bases de datos tanto de forma local como remota. Es posible acceder a servidores Oracle a través del BDE (Borland Data Engine).

Kylix produce directamente código ejecutable, de tal forma que las aplicaciones no dependen de otros programas o librerías.

Debido a problemas con algunas de la librerías comunes de Linux (lo cual significa requisitos de Kylix sin cumplir) es imprescindible ejecutar un programa que viene con kylix, antes de instalarlo. Este programa chequea si las librerías críticas corresponden con una nueva versión adecuada y así no sufrir después los fallos.

Asumiendo que el punto de montaje es:

```
/home/pablo/entornos_de_desarrollo/kylix/kylix3_open
```

Esto ejecutará el test de pre instalación:

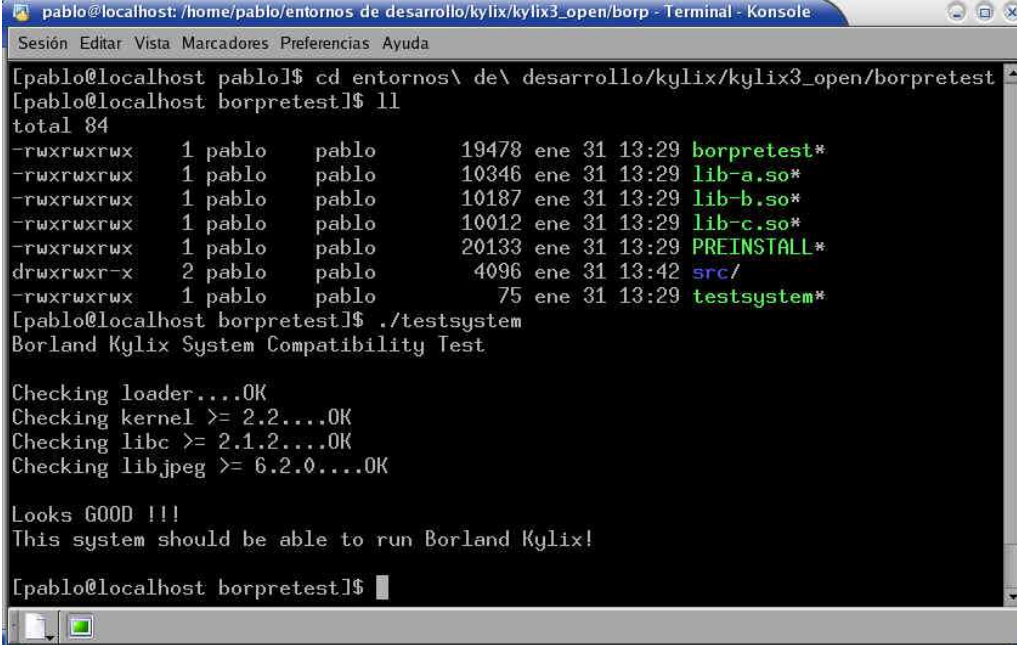
```
cd /home/pablo/entornos_de_desarrollo/kylix/kylix3_open/borpretest
```

```
./testsystem
```

En la figura 1, se puede ver la salida del programa en una distribución válida. Sin embargo, si el programa dice que una de esas librerías no vale se tiene que actualizar.

El fichero *preinstall* explica el procedimiento de actualización de la librería *jpeg* y *glibc* de las distribuciones soportadas. Para otras distribuciones se necesita encontrar e instalar librerías actualizadas.

Figura 1: Programa de test de compatibilidad de una distribución válida



```
pablo@localhost: /home/pablo/entornos de desarrollo/kylix/kylix3_open/borp - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
[pablo@localhost pablo]$ cd entornos\ de\ desarrollo/kylix/kylix3_open/borpretest
[pablo@localhost borpretest]$ ll
total 84
-rwxrwxrwx  1 pablo  pablo    19478 ene 31 13:29 borpretest*
-rwxrwxrwx  1 pablo  pablo    10346 ene 31 13:29 lib-a.so*
-rwxrwxrwx  1 pablo  pablo    10187 ene 31 13:29 lib-b.so*
-rwxrwxrwx  1 pablo  pablo    10012 ene 31 13:29 lib-c.so*
-rwxrwxrwx  1 pablo  pablo    20133 ene 31 13:29 PREINSTALL*
drwxrwxr-x  2 pablo  pablo     4096 ene 31 13:42 src/
-rwxrwxrwx  1 pablo  pablo      75 ene 31 13:29 testsystem*
[pablo@localhost borpretest]$ ./testsystem
Borland Kylix System Compatibility Test

Checking loader...OK
Checking kernel >= 2.2....OK
Checking libc >= 2.1.2....OK
Checking libjpeg >= 6.2.0....OK

Looks GOOD !!!
This system should be able to run Borland Kylix!

[pablo@localhost borpretest]$
```

Una vez se aceptan las librerías del sistema, ya se puede proceder a instalar kylix:

1. Se descomprimen los archivos `kylix3_open.tar.gz` y `fix_kylix3.tar.gz`. Utilizando la opción "extraer aquí" que se visualiza en el menú emergente del ratón (clic derecho sobre el icono). O por consola de comandos se digita lo siguiente:

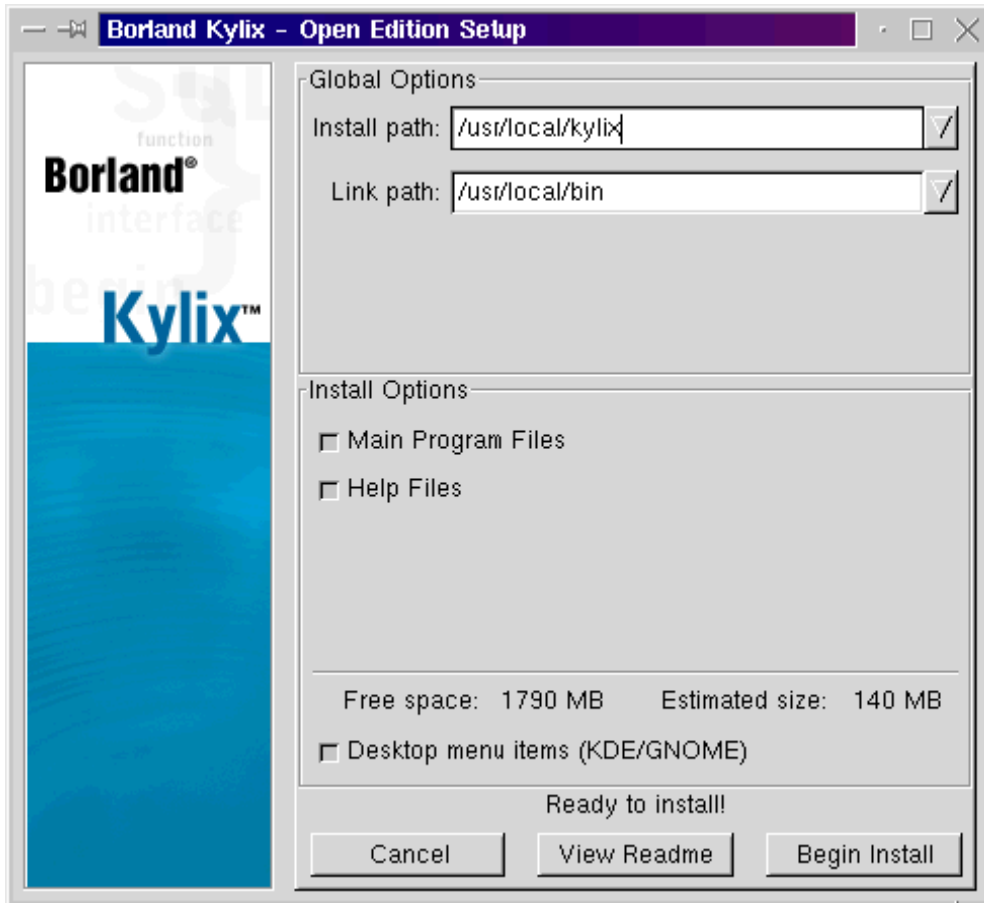
```
tar zxvf kylix3_open.tar.gz
tar zxvf fix_kylix3.tar.gz
```

2. Una vez descomprimidos y desempaquetados los archivos, ingresamos a la carpeta que se crea (`kylix3_open`) la cual aparece en el lugar donde se hizo el paso anterior y se digita el siguiente comando (en una terminal o consola)

```
sh setup.sh
```

3. Acepte el contrato de licencia
4. Si se tiene las librerías `GTK` en la máquina, el programa `setup` ejecuta una aplicación X (ver Figura 2), en caso contrario ejecuta simplemente la parte de consola. El fichero `install` muestra la mejor forma de instalar el software como `root`, pero que no se instale en el directorio `root` (instálelo en `/opt/kylix` o `/usr/local/kylix`).

Figura 2: Listo para instalar kyllix



5. Para registrar el producto simplemente copie el archivo `reg92.txt` y péquelo en la carpeta que se creo de `kylix3`.

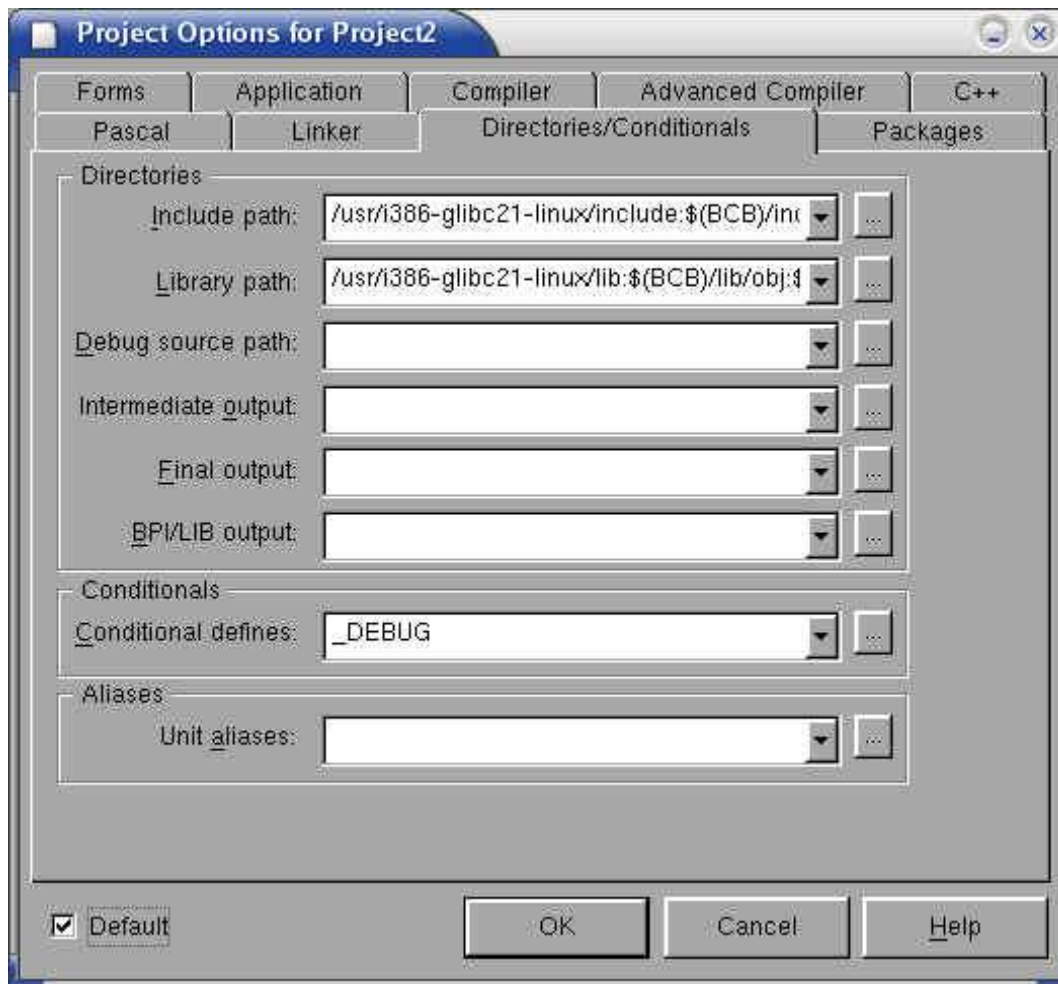
6. Haga doble clic en el archivo `compat_glibc7_2_2_1_3mdk.rpm` o en la consola escribiendo:

```
rpm -Uvh compat_glibc_7_2_2_1_3mdk.rpm
```

o lo que salga.

7. En la carpeta `kylix3` haga doble clic en `starbcb` para entrar a `kylix`, luego entran en Opciones del proyecto (ver figura 3), luego en directorios y agregan en el `include path` el directorio `/usr/i386-glibc21-linux/include`, después se tiene que subir este directorio al principio de la lista. Se hace lo mismo para el `library path` pero agregando el directorio `/usr/i386-glibc21-linux/lib`

Figura 3: Opciones del proyecto



8. Por último se hace clic en *default* para que no se tenga que hacer este mismo proceso siempre.

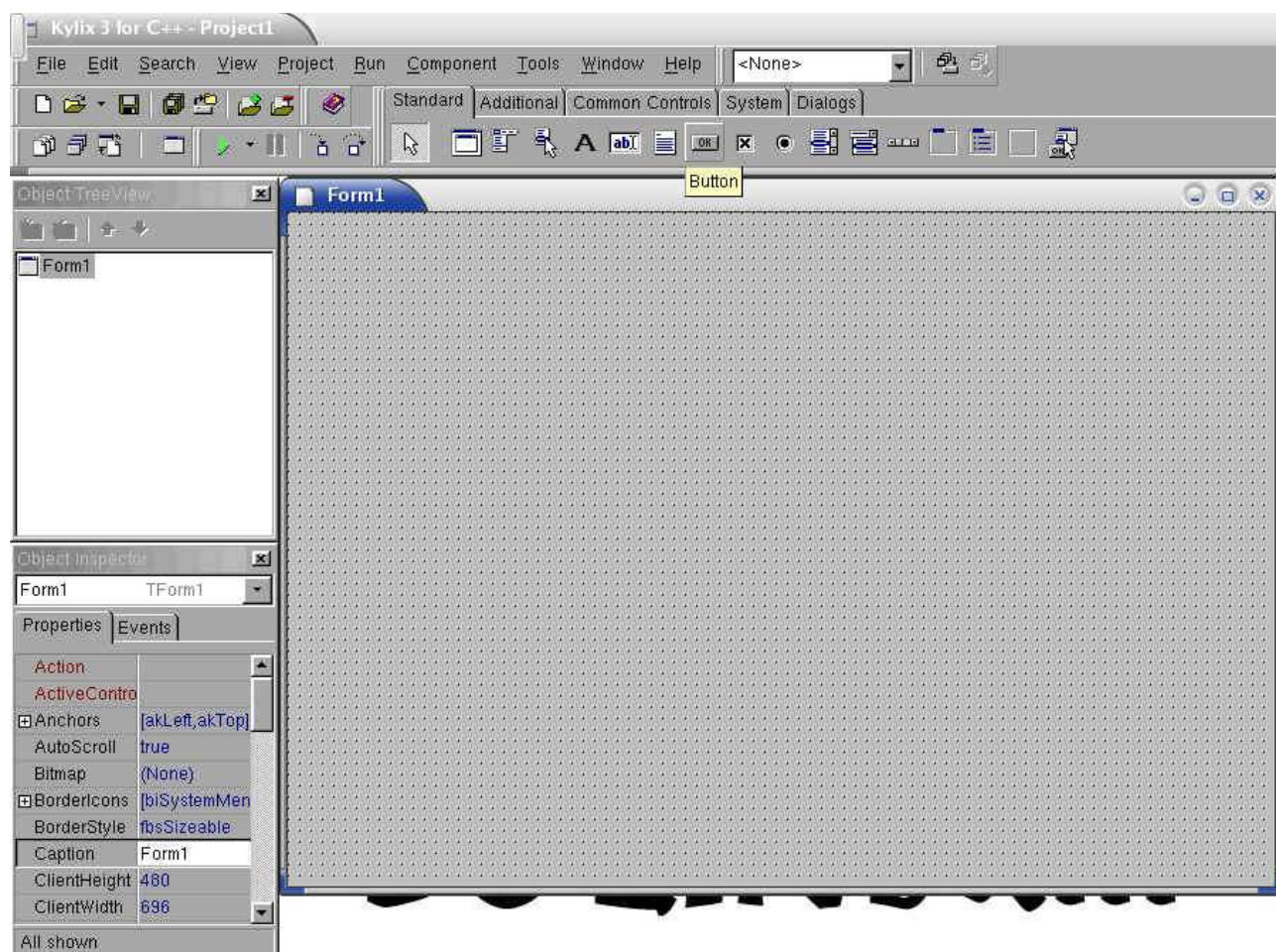
Es necesario que el *link path* sea algún directorio del *path* del sistema, ya que es donde se almacenarán algunos *scripts* que serán invocados por Kylix, el sistema de ayuda de Kylix y también el compilador de línea de comandos.

Una vez esté Kylix instalado, se tiene que reiniciar las X para que aparezca en el menú los elementos necesarios para lanzarlo correctamente, y asegurarse de entrar como un usuario normal. Se puede encontrar más información de utilidad sobre Kylix en el fichero *README* e información de uso de tus aplicaciones en Kylix en *DEPLOY*.

La primera vez que se ejecuta Kylix aparece una ventana blanca en la pantalla durante un rato avisando que está generando la métrica de fuentes. Se puedes

ignorar esta ventana sin problemas, generalmente no volverá a aparecer. Una vez hecho esto, el Entorno de Desarrollo Integrado (IDE) de Kylix aparecerá en pantalla (ver Figura 4).

Figura 4: El IDE de Kylix



La ventana de abajo (Figura 5) realmente no tiene nombre (se puede llamar ventana principal). La ventana principal contiene los menús más importantes, así como gran cantidad de barras de herramientas personalizables que tienen gran variedad de botones, que sirven como accesos directos a elementos del menú.

El elemento que queda en la ventana principal es el área de pestañas llamada *Paleta de Componentes*, muestra todos los componentes disponibles para diseñar formularios en las aplicaciones.

Figura 5: Ventana principal

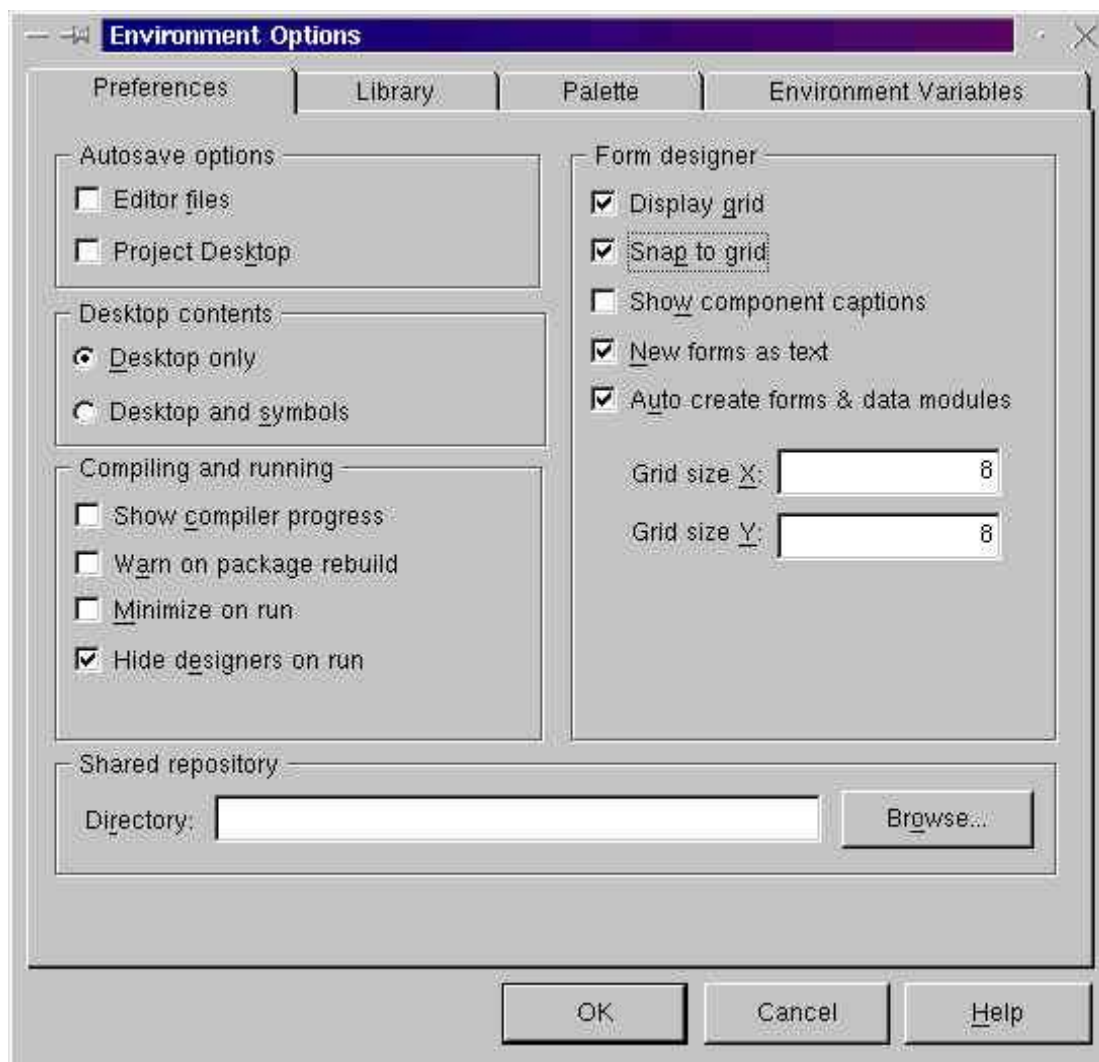


Un componente es un objeto de un programa, como por ejemplo un botón o un control de edición. Si se pasa con el ratón por encima de las imágenes de la *Paleta de Componentes*, aparecerán etiquetas diciendo qué es cada cosa. Cada componente tiene un tipo de dato implícito. Por ejemplo el componente *button* es tipo *TButton*, y tiene una etiqueta que dice *Button*; así como el componente *checkbox* es de tipo *TCheckBox* y tiene un etiqueta que pone *Checkbutton*.

Kylix viene con una librería de componentes llamada *CLX (Librería de Componentes Multiplataforma)*. Estos componentes se agrupan en cuatro solapas o páginas de componentes.

Un formulario es el nombre dado para designar tanto a una ventana como a un cuadro de diálogo. Todos los formularios se diseñan del mismo modo. A no ser que se esté diseñando una aplicación para consola, la mayoría de las aplicaciones Kylix tendrán al menos un formulario. Cada formulario se diseña usando el *Diseñador de formularios*. Se puede ver un formulario en blanco (es decir, sin ningún componente sobre él) en la captura de pantalla del IDE. Es una ventana bastante grande con una leyenda que pone *Form1* y rellena de puntos, los cuales indican la posición de una rejilla sobre la que se acoplan los componentes que se van soltando en el formulario, para ayudarlos a mantener alineados. Se puede configurar el *Diseñador de Formularios*, cambiando la separación de la rejilla, si está activa o visible en la página *Preferences* del diálogo *Environment Options* (ver figura 6), que se encuentra en el menú *Tools*.

Figura 6: Cambia aquí las opciones de Kylix



Para colocar componentes sobre un formulario, clic sobre la imagen del componente en la paleta de componentes y luego vuelva a hacer clic sobre el *Diseñador de Formularios* y podrá mover o redimensionar según necesite. Se pueden colocar todos los componentes que necesite (ver Figura 7).

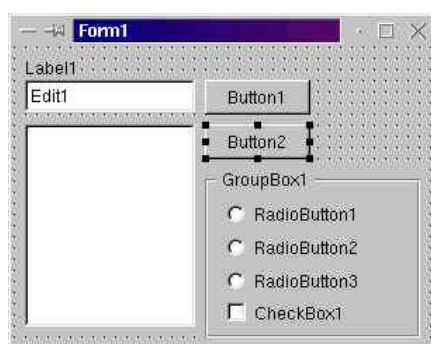


Figura 7: El diseñador de Formularios con algunos componentes

El *Inspector de Objetos* se puede ver en esta captura de pantalla del IDE, a la izquierda del Diseñador de Formularios. El Inspector de objetos muestra las propiedades del componente seleccionado y permite modificar

sus valores iniciales. Esto permite diseñar rápidamente la apariencia inicial de todos los formularios. Por ejemplo, se puede modificar la leyenda de un botón con su propiedad `Caption`.

Para seleccionar un componente del *Diseñador de Formularios*, simplemente se hace click sobre él, esto permitirá moverlo, redimensionar o modificarlo (ver Figura 8).

Además de poder modificar las propiedades, el *Inspector de Objetos* también permite saber el nombre y el tipo del componente. La lista de instancias en la parte superior del inspector de objetos (se llama así porque muestra una lista de todas las instancias de componentes del diseñador de formularios y permite seleccionarlos) muestra el tipo de componente. El nombre del componente también viene indicado más abajo como valor de la propiedad `Name`. Se puede cambiar el nombre del componente igual que las demás propiedades.

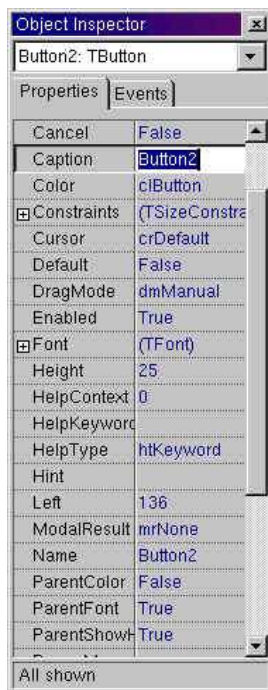


Figura 8: Propiedades de Button

El inspector de objetos tiene dos solapas, una es *Properties*. La otra es *Events* (Ver Figura 9) muestra todos los eventos que podrán ocurrir el seleccionar el componente. Este es el principal mecanismo para construir aplicaciones Kylix, se escribe código que se ejecutará en respuesta a los eventos del componente (manejadores de eventos). Por ejemplo, cuando un usuario presiona un componente `TButton`, se llama al evento `OnClick`. El manejador de evento `OnClick` para el botón puede mostrar un formulario, lanzar otra aplicación, terminar la aplicación y cualquier otra cosa.

Para hacer manejadores de eventos hay que hacer doble click en la columna del lado derecho del evento en cuestión (o seleccionando el evento en el inspector de objetos y pulsando `ctrl+enter`). El código se escribe en la otra ventana que Kylix abre por defecto al arrancar (ver Figura 9) que muestra un manejador de evento para el evento `OnClick` del `Button1`. Normalmente está detrás del diseñador de formularios. El editor puede tener muchos ficheros abiertos simultáneamente, y también es posible tener múltiples editores abiertos (*View|New Edit Window*).

El desarrollo de aplicaciones visuales en Kylix consiste en configurar componentes sobre formularios y luego escribir los manejadores para aquellos componentes y formularios que se necesiten. Los manejadores de eventos pueden manipular los objetos en tiempo de diseño, acceder a sus propiedades o llamar a otros métodos o pueden hacer uso de cualquier otra parte de la

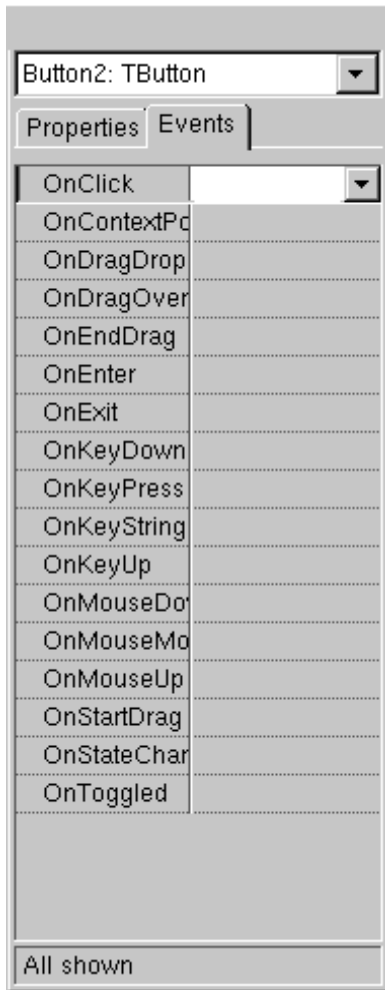


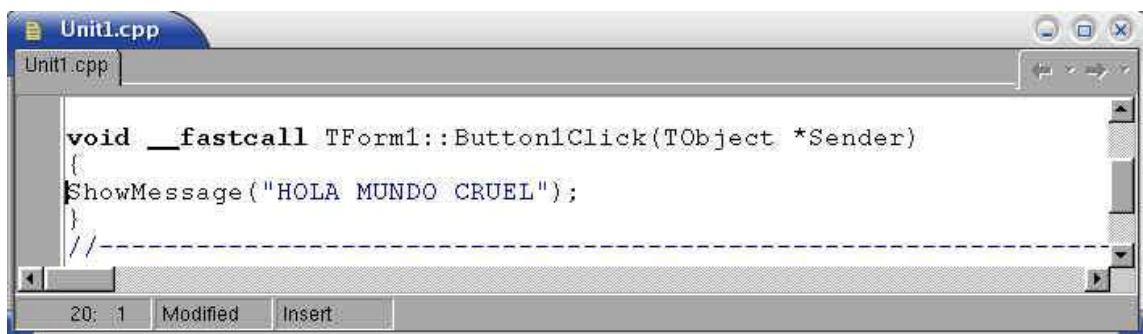
Figura 9: Eventos de Button

librería *CLX*. En algunas ocasiones se dice que las herramientas de desarrollo de Kylix siguen un modelo de componente *PME*, debido a que los componentes son un conjunto de Propiedades, Métodos y Eventos.

Para empezar un proyecto nuevo para una aplicación, elegir *File | New Application*.

Coloque un componente *button* sobre el formulario y use el *Object Inspector* para modificar la propiedad *caption* a: "HOLA MUNDO CRUEL". Ahora cambie la solapa *events* del *Object Inspector* y localiza el evento *onclick*, que es el que se llama cuando se hace clic sobre el botón. Haga doble click en la columna de la derecha (o seleccione el evento y pulse *ctrl+enter*) para crear un manejador de evento para dicho evento (ver Figura 10). En la línea en blanco que aparece escriba lo que aparece en la figura 10:

Figura 10.



ShowMessage es una sencilla rutina *CLX* que produce un dialogo. Para probar esta aplicación, elija *Run* o pulse *F9* (o el botón con un triangulo verde). Kylix compilará el código fuente, lo enlazará en un ejecutable y lo ejecutará. Cuando se pulsa el botón, un cuadro de mensaje aparece diciendo "HOLA MUNDO

CRUEL" (ver figura 11)



Figura 11: primera aplicación en Kylix

Al inicio del programa, el *banner* de Kylix Open Edition aparece como un modal, ventana siempre visible y sin borde, durante aproximadamente 3 segundos (ver figura 12), haciendo clic sobre ella desaparece y pasándole el parámetro `-ns` por línea de comandos a nuestra aplicación no vuelve a aparecer.

Figura 12: Banner de Kylix Open Edition

