

### 3.1. wxApp, wxFrame y wxButton

#### wxApp

Todas las aplicaciones de wxWidgets están definidas en clases derivadas de la clase wxApp. Una aplicación en ejecución no es más que una instancia de ella.

Como ya hemos visto en el capítulo anterior, cada clase que definamos derivada de wxApp debe de tener un método *OnInit()* que será llamado cuando wxWidgets esté listo para ejecutar el código. Sería equivalente a las funciones *main()* de C o *WinMain()* de Win32.

Aquí tenemos un ejemplo de como se definiría una clase desde wxApp:

```
class MyApp : public wxApp
{
    public:
        virtual bool OnInit();
};
```

La implementación del método *OnInit()* contendrá las sentencias que deseamos que se ejecuten al cargar la aplicación, o sea, lo que podríamos llamar "la configuración inicial" de la aplicación. Esta función devolverá un valor booleano en función de si se ha ejecutado correctamente o no.

Entonces, ¿en qué parte del código debemos decir que creamos una instancia de esta clase? wxWidgets la crea internamente, pero nosotros debemos decirle que clase de objeto deseamos crear. Por lo que debemos utilizar una macro en nuestro fichero de código, **IMPLEMENT\_APP(nombre\_clase)**.

Cuando wxWidgets crea un objeto aplicación derivado de wxApp se le asigna el resultado a la variable global wxTheApp. Esta se puede usar en la aplicación, pero no es conveniente. Para ello podemos insertar la macro **DECLARE\_APP(nombre\_clase)** después de la definición de la clase en el fichero de cabecera. Esto nos permite usar la función *wxGetApp()* que nos devuelve una referencia al objeto aplicación que hemos creado.

#### wxFrame

Es una ventana de bajo nivel, una clase derivada de wxWindow. Puede contener otras ventanas y suele tener barra de menú y barra de título.

Su definición sería de la siguiente manera:

```
class MyFrame : public wxFrame
{
    public:
        //Constructor.
        //Manejadores de eventos.
    private:
        DECLARE_EVENT_TABLE()
};
```

El prototipo del constructor es:

```
wxFrame(wxWindow* parent, wxWindowID id, const wxString& title,
```

```
const wxPoint& position, const wxSize& size,
long style, const wxString& name);
```

Los eventos que genera esta clase son los siguientes: **EVT\_ACTIVATE**, cuando se activa o desactiva el `wxFrame`, **EVT\_CLOSE**, cuando se intenta cerrar el `wxFrame` (ya sea el usuario o el sistema), **EVT\_ICONIZE**, cuando se intenta minimizar o restaurar, y **EVT\_MAXIMIZE**, cuando se maximiza.

Algunos de los principales métodos de la clase `wxFrame` son: *CreateStatusBar()*, *SetStatusText()* y *SetStatusWidths()* para gestionar la barra de estado, *CreateToolBar()*, *GetMenuBar()* y *SetMenuBar()* para gestionar barras de herramientas, *GetTitle()* y *SetTitle()* para ver y editar el título, *Iconize()*, *IsIconized*, *Maximize()* e *IsMaximized* para comprobar y modificar el estado de la ventana, *SetIcon()* establece el icono de la ventana y *ShowFullScreen()* muestra el componente al tamaño máximo posible.

## **wxButton**

Este componente es un control representado como un botón pulsable con una etiqueta de texto

Para crearlo debemos pasarle al constructor la ventana padre, el identificador, la etiqueta, la posición, el tamaño y el estilo.

```
wxButton(wxWindow* parent, wxWindowID id,
const wxString& label = wxEmptyString,
const wxPoint& pos = wxDefaultPosition,
const wxSize& size = wxDefaultSize, long style = 0)
```

Este componente cuenta con numerosos estilos: **wxBU\_LEFT**, **wxBU\_RIGHT**, **wxBU\_TOP**, **wxBU\_BOTTOM**, para alinear el texto respecto al botón, **wxBU\_EXACTFIT**, que crea el botón de tamaño mínimo para que entre el texto, y **wxNO\_BORDER**, que crea un botón sin bordes.

El único evento que genera este componente es **EVT\_BUTTON**, cuando se pulsa el botón.

Este componente tiene pocos métodos, *SetLabel()* y *GetLabel()*, para ver y modificar la etiqueta, y *SetDefault()*, para restaurar el botón a uno predeterminado.

Además, este componente cuenta con las llamadas etiquetas predeterminadas, que son una serie de macros ya definidas que podemos utilizar como identificador del botón, modificando automáticamente la etiqueta del mismo. A continuación se muestran unas cuantas: **wxID\_ADD**, **wxID\_CANCEL**, **wxID\_CLOSE**, **wxID\_DELETE**, **wxID\_NO**, **wxID\_OK**, **wxID\_EXIT**, **wxID\_SAVE**, **wxID\_YES**, etc.