

Programación Orientada a Objetos

Unidad 9: GUIs

DESARROLLO DE PROYECTOS CON GUIs

- ▶ Paso 1: **nada de GUIs**

- ▶ Modelar el problema

- ▶ Armar una "biblioteca" de clases y funciones que modelen mi problema de forma independiente de la interfaz

- ▶ Paso 2: **nada de GUIs**

- ▶ Probar el modelo

- ▶ Asegurarse de que las clases y funciones andan y tienen todo lo necesario

- ▶ Paso 3: Por fin! Hacer la GUI

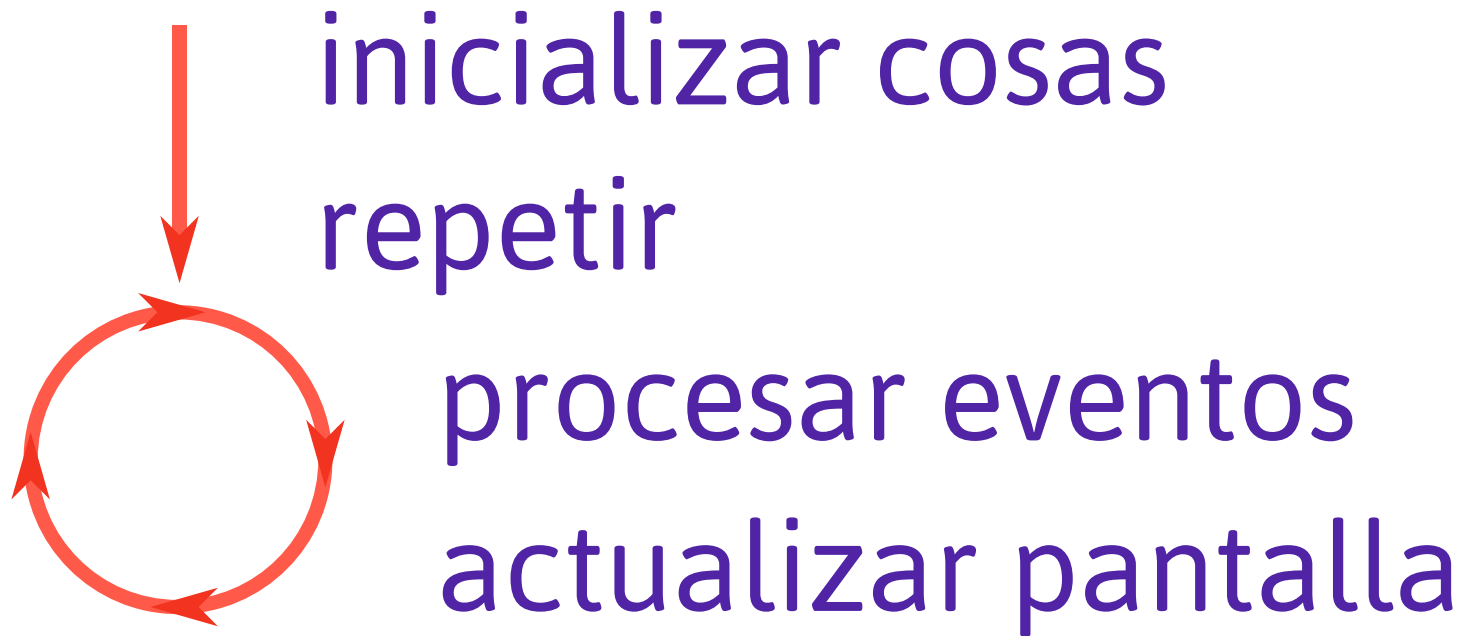
¿EVENTOS?

En programación, un evento es un suceso "interesante" para una aplicación

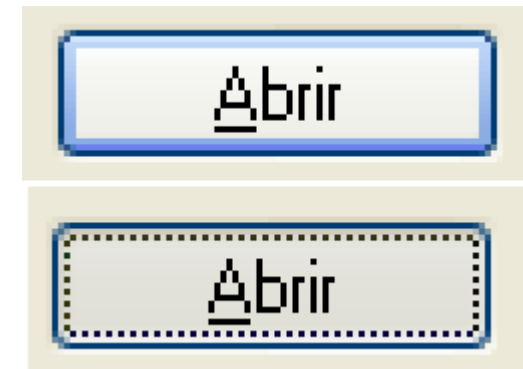
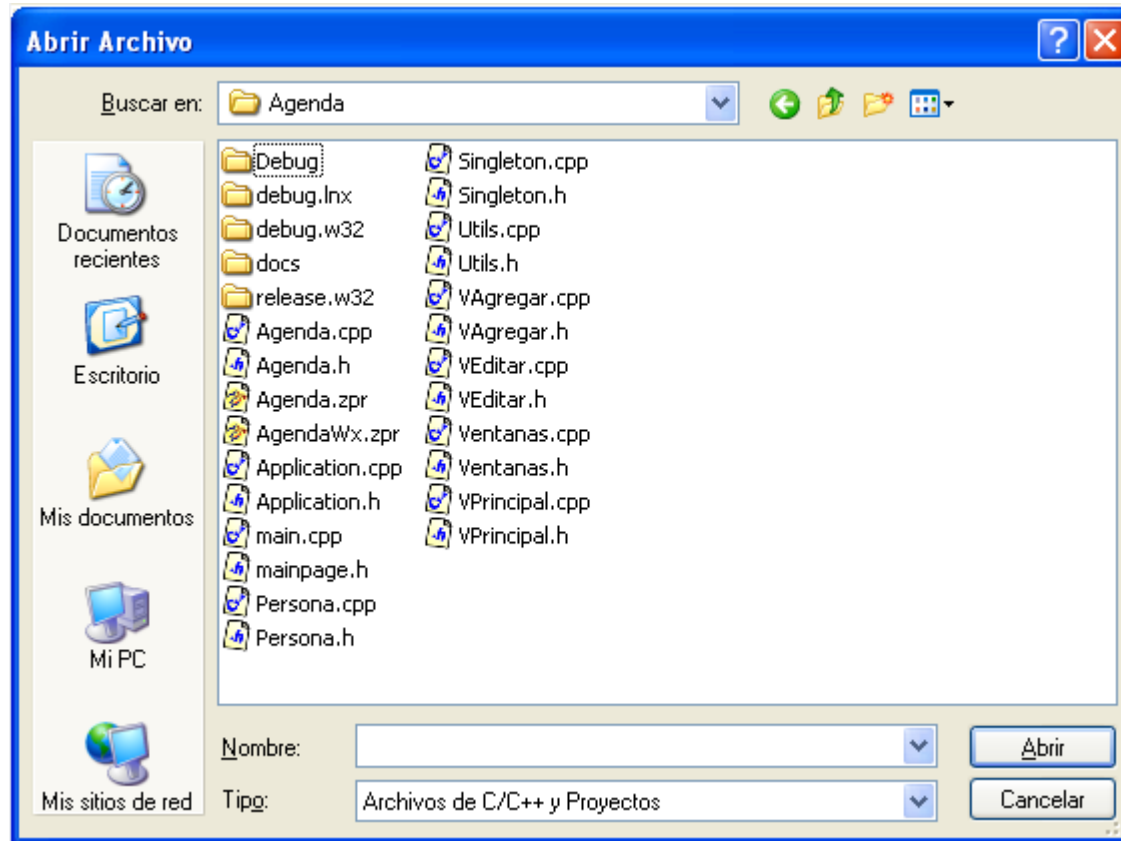
Ejemplos (en GUIs):

- ▶ Se presiona una tecla
- ▶ Se hace click sobre algún botón
- ▶ Cambia de tamaño la ventana
- ▶ Se elige una opción de un menú
- ▶ Se edita el contenido de un cuadro de texto
- ▶ Se presiona un atajo de teclado
- ▶ El usuario intenta cerrar la ventana
- ▶ ...

LOOP DE EVENTOS



EJEMPLO: DIÁLOGO ABRIR

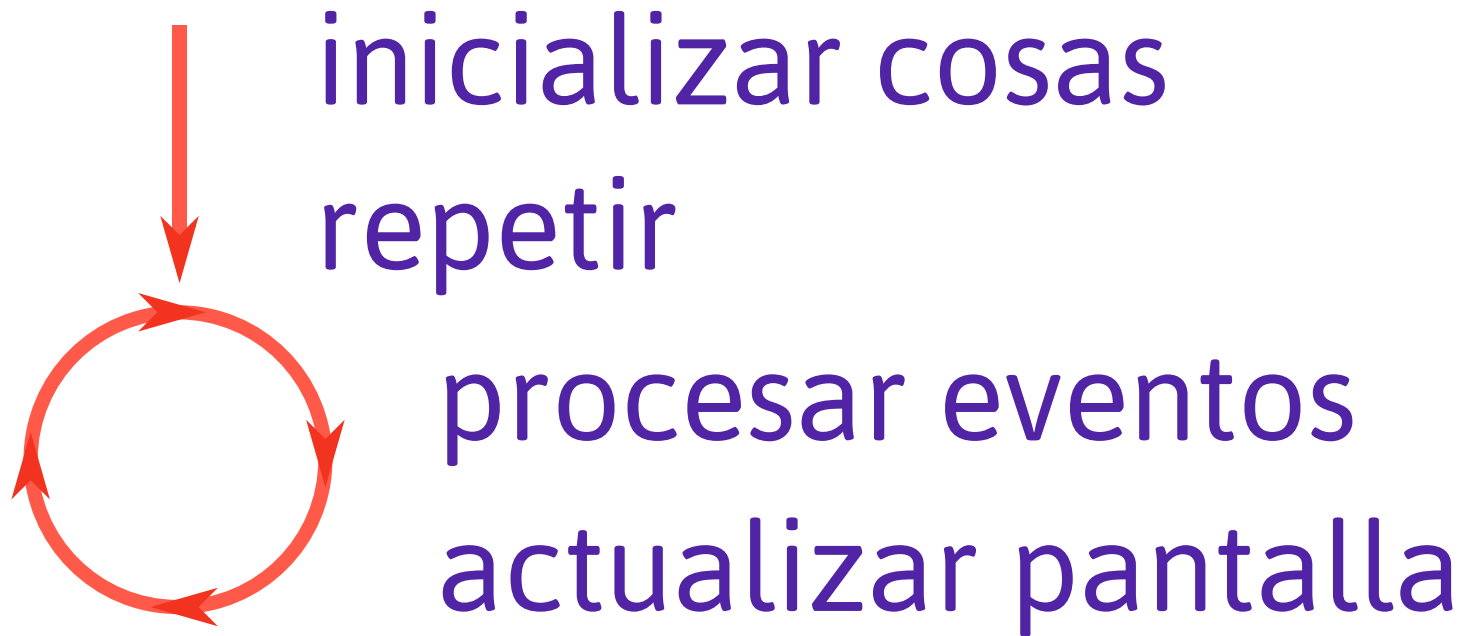


¿color?
¿tipografía?
¿posición?
¿tamaño?
¿etc...?

EJEMPLO: DIÁLOGO ABRIR

1. Observar que se presiona el botón del ratón
2. Detectar que el puntero estaba sobre el botón
3. Actualizar dibujo del botón en la pantalla
4. Observar que se suelta el botón del ratón
5. Actualizar dibujo del botón en la pantalla
6. Mandar a procesar el evento (abrir el archivo)

LOOP DE EVENTOS



❓ ¿Quién y cómo lo implementa?

INVERSIÓN DEL CONTROL Y PRINCIPIO DE HOLLYWOOD

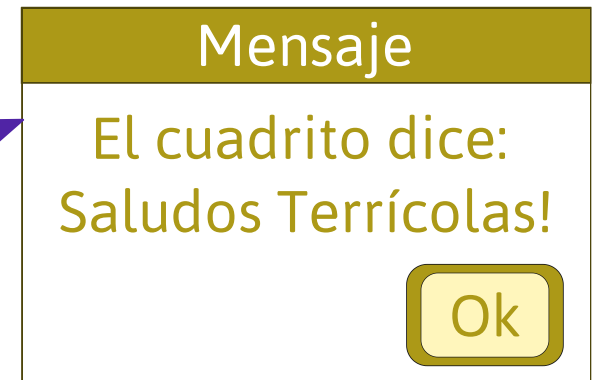
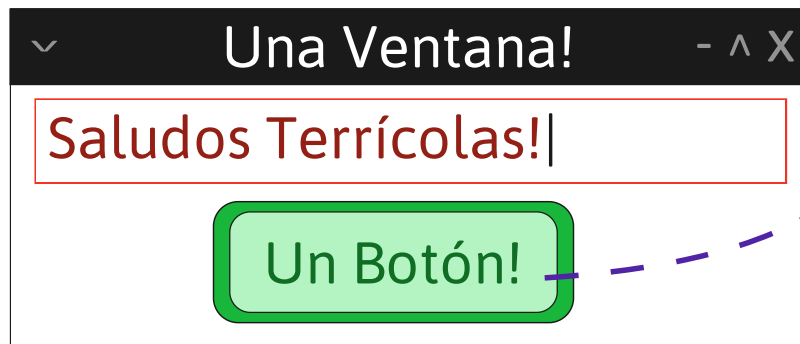


EVENTOS Y OBJETOS CON GUIs DE ALTO NIVEL

- ▶ Las ventanas son clases:
 - ▶ Los controles son sus atributos.
 - ▶ Los eventos son métodos.
- ▶ Crear una ventana es instanciar una clase.
 - ▶ Los controles se agregan por composición.
- ▶ Obtener o setear una propiedad de un control es invocar un método de un objeto atributo.
- ▶ El comportamiento para un evento se define implementando un método.

EJEMPLO

```
class MiVentana : public Ventana {  
    Texto *txt;  
    Boton *btn;  
public:  
    MiVentana() : Ventana("Una Ventana!") {  
        txt = new Texto(this, "texto inicial");  
        btn = new Boton(this, "Un Boton!");  
        btn->Conectar(CLICK, MiVentana::FuncionBoton);  
    }  
    void FuncionBoton() {  
        string msg = "El cuadrilo dice:\n";  
        MostrarMensaje(msg + txt->VerTexto());  
    }  
};
```

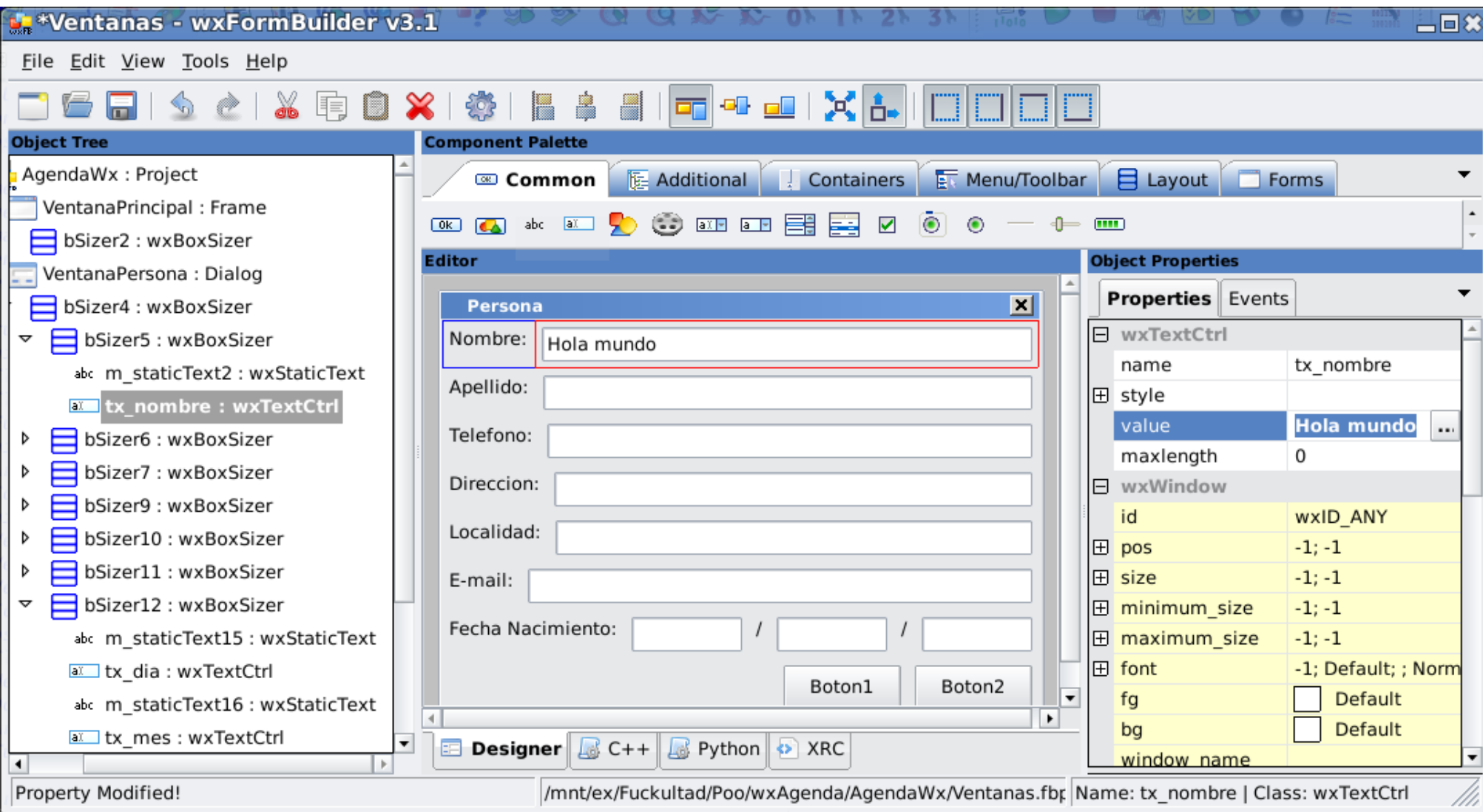


CASO DE ESTUDIO: WXWIDGETS

Algunas ventajas:

- Gran variedad de controles y ventanas
- No solo controles visuales (threads, strings archivos, conexiones de red, etc)
- Portable (Windows, Linux, Mac OS, etc)
- Gratuita y de código abierto (mejor que GPL)

WXFORMBUILDER



WXFORMBUILDER

En el diseñador se **dibujan** las ventana
y se **eligen** los eventos interesantes.

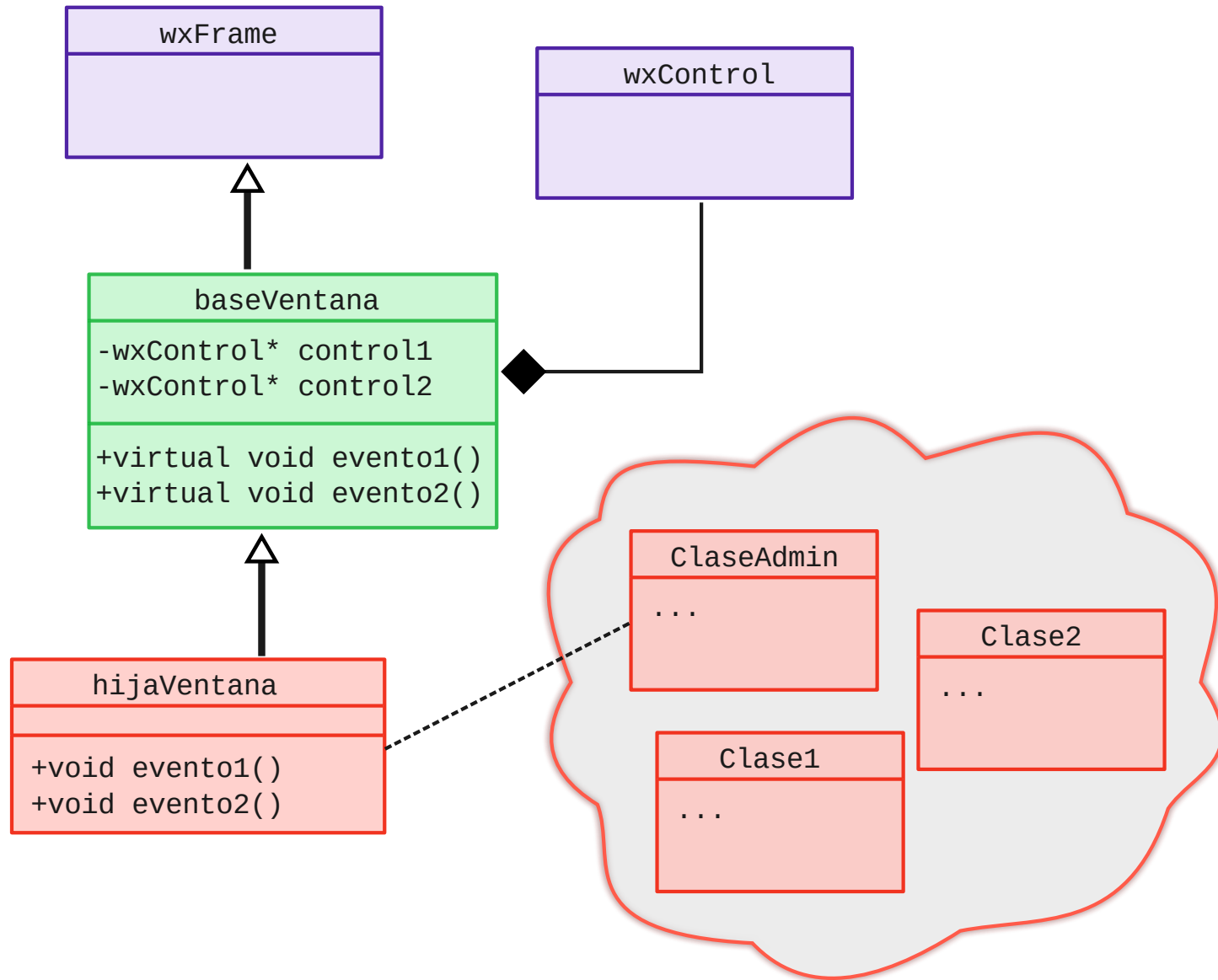
El diseñador genera:

- ▶ Una clase por ventana
- ▶ Los controles serán atributos (punteros)
- ▶ El constructor crea todos los controles
- ▶ Los eventos serán métodos virtuales

❓ ¿por qué virtuales?

Hay que **heredar** nuevas clases desde las que
genera wxfb para reimplementar sus Métodos
virtuales

¿QUIÉN IMPLEMENTA QUÉ?



¿QUÉ HACE QUIÉN?

Zinjal

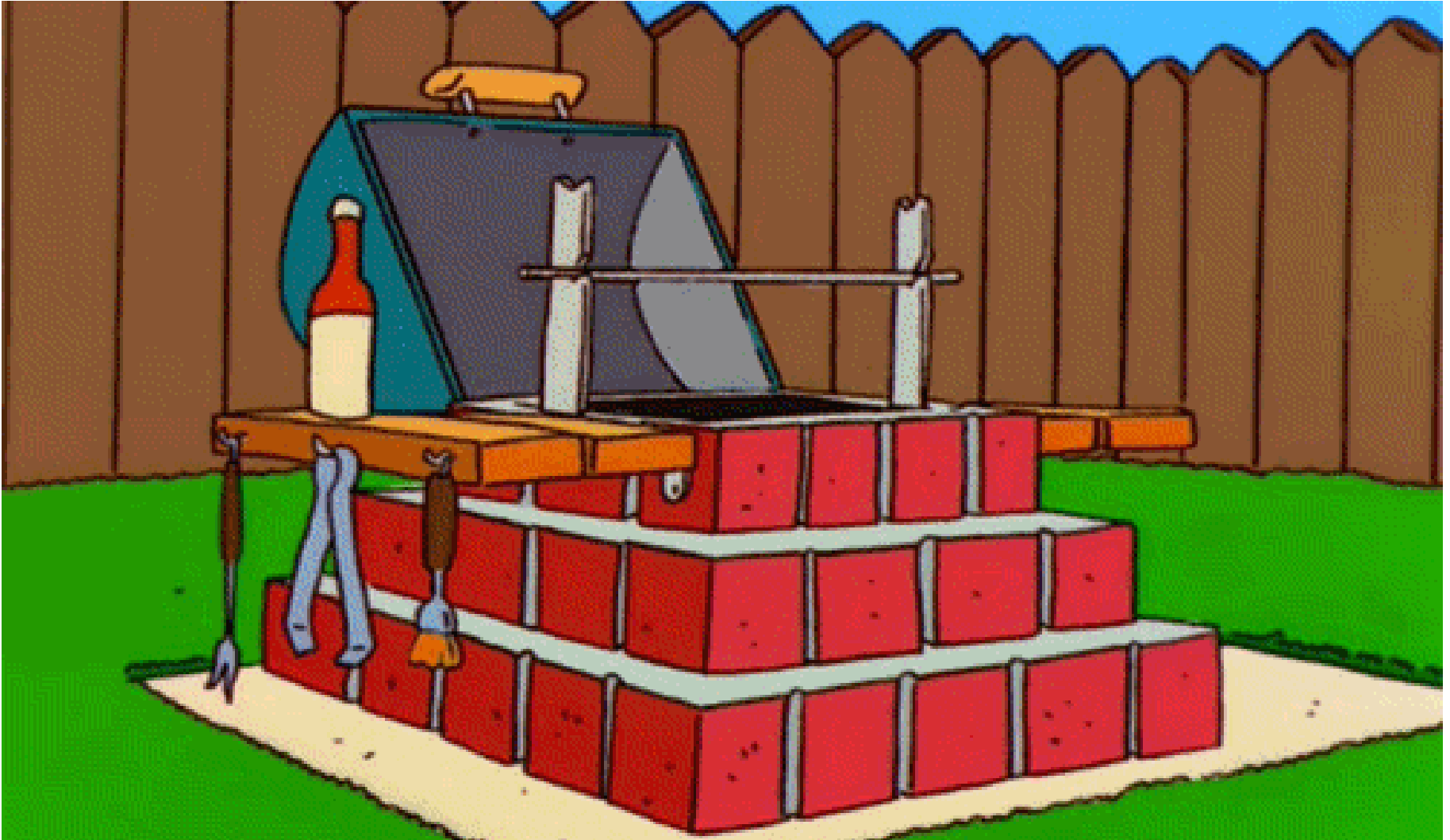
crear un proyecto wxfb
en blanco y configurar la
compilación con wxWidgets



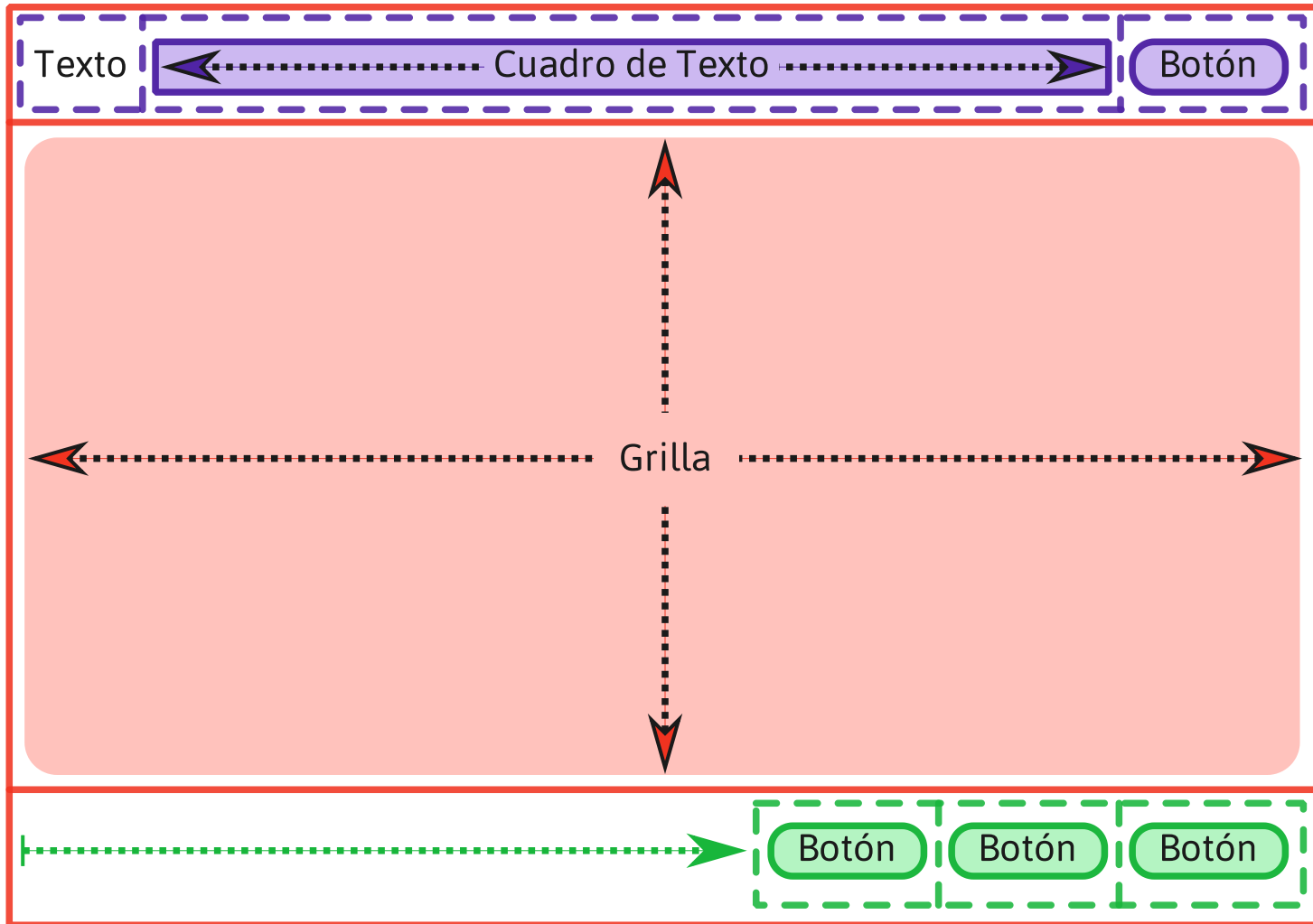
dibujar las ventanas y
elegir cuales serán los
eventos interesantes

Tú

ORGANIZACIÓN DE LOS CONTROLES EN UNA VENTANA...



ORGANIZACIÓN DE LOS CONTROLES EN UNA VENTANA...



- ⚠ Siempre es mejor evitar valores absolutos
- ✅ Lo más simple es combinar objetos de tipo `wxBoxSizer`

¿QUÉ HACE QUIÉN?

