



Programación orientada a Eventos

Taller de Programación I - Cátedra Veiga - FIUBA



Repasemos otros paradigmas conocidos

- Prog. procedural: Procedimientos imperativos
- POO: objetos (estado + comportamiento) y sus relaciones
- Prog. funcional: No hay estado mutable



POE

La ejecución del programa está determinada por **eventos** del entorno.

Es el paradigma clave en el centro de todos los frameworks de GUIs.



Algunos tipos de eventos

- Teclado, mouse
- Recepción de mensajes de otro programa
- Paso del tiempo



Acciones

Una **acción** es la respuesta a un evento, también conocido como **event handler**.

Event handler: función que está *atada* (**bind**) al evento, y se encarga de manejarlo.

Los eventos pueden o no tener acciones definidas.



Event Queue

Los eventos generados serán pusheados a una **event queue** para ser manejados.



Event Loop

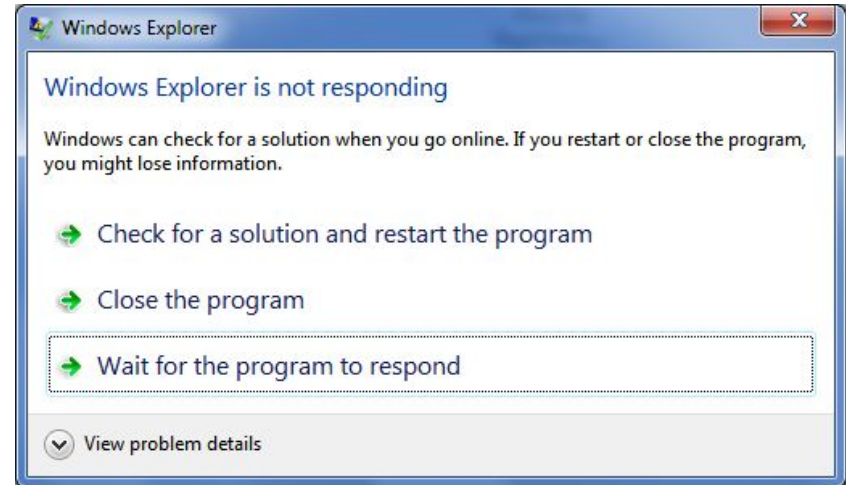
- Se compone de una event queue.
- Es fundamental que **no se bloquee**.



```
while (true) {  
    event = event_queue.pop();  
  
    if (event == quit) {  
        break;  
    }  
  
    handler = search_handler_for(event);  
  
    if (handler){  
        handler();  
    }  
}
```

¿Por qué es fundamental que no se bloquee?

Si un manejador se bloquea, no se procesarán más eventos.





¿Cómo evitamos que se bloquee?

- Handlers rápidos
- Tareas pesadas -> offload a un hilo en background (¿cómo joineamos ese hilo?)
- Dividir la tarea en subtareas y hacer polling de los estados intermedios (asincronismo)