

# ***Programación Avanzada***

## **IIC2233 2023-2**

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Joaquín Tagle - Francisca Cattán



# Anuncios

1. Hoy tenemos la segunda experiencia.
2. Encuesta de Carga Académica y Evaluación Temprana de Cursos.  
¡Respóndanlas!

---

# Interfaces gráficas

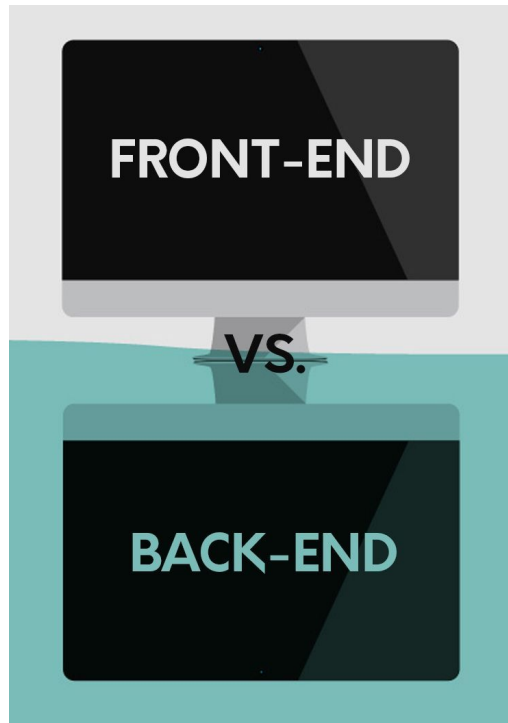
# Patrón de diseño *front-end/back-end*

¿En qué consiste?

La separación que existe entre la capa de **presentación** (*front-end*) y la capa de **acceso a los datos** (*back-end*).

¿Por qué tenemos esto?

Para asegurar un **bajo acoplamiento** y una **alta cohesión**.



# Interacción

*front-end*  
*back-end*

- Entradas
- Salidas
- Control de la aplicación

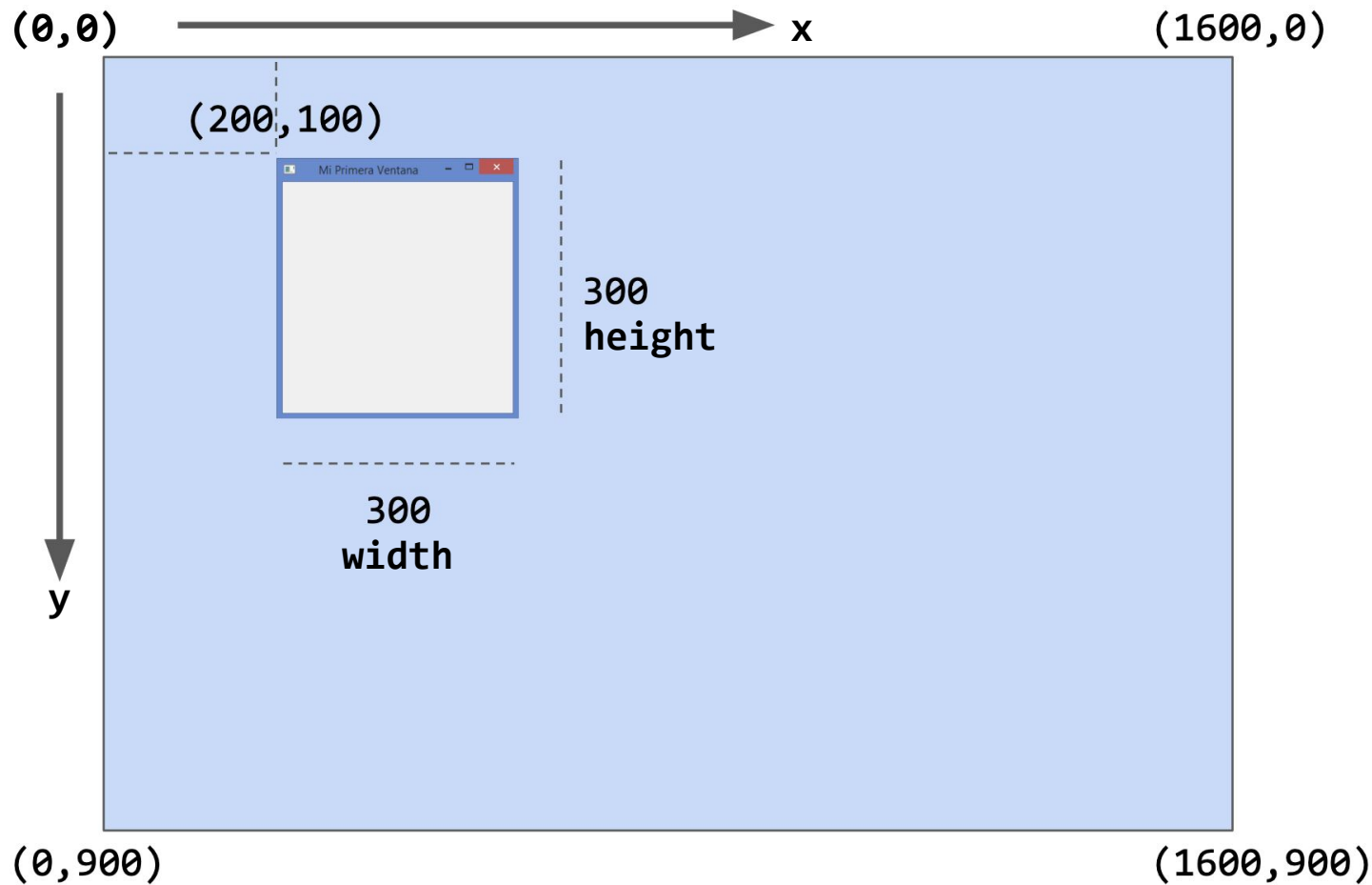
---

***Front-end***

# ***Widgets***

- Etiquetas (*Labels*)
- Cuadros de texto (*LineEdits*)
- Botones (*PushButtons*)
- Imágenes (*PixMaps*)
- Disposiciones (*Layouts*)
- y muchos más

---





***Back-end***

# Componentes lógicos

- Se encargan de procesar toda la información lógica del programa.
- Ayudan a mantener ciertos estados y su actualización.
- Permite la comunicación entre los distintos componentes visuales.

---

# Comunicación entre *front-end* y *back-end*

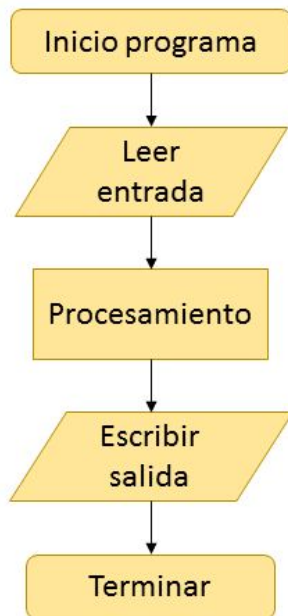
# Eventos

**Fuente**  
*(sender)*

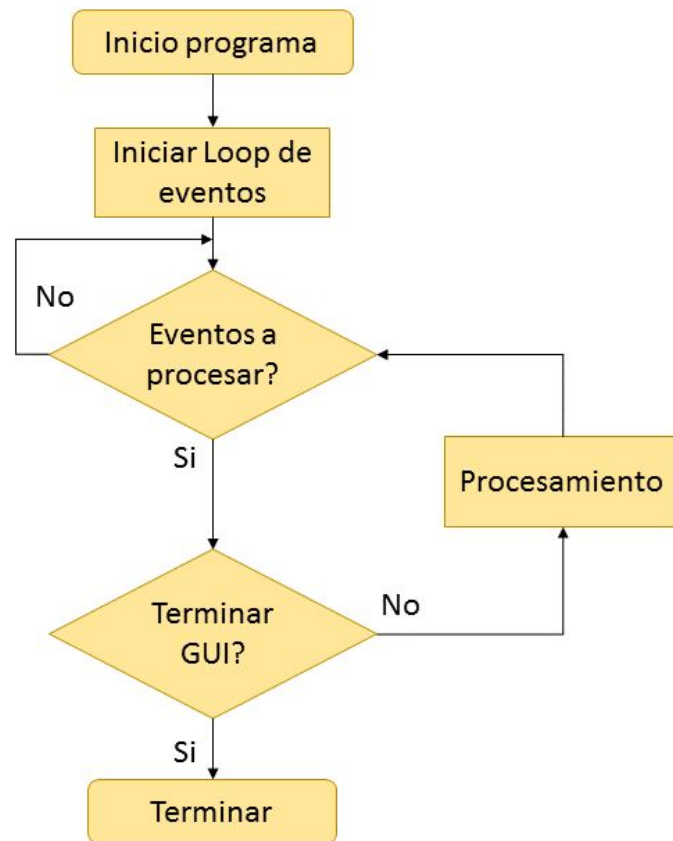


**Señal**

**Objetivo**  
*(recipient)*



a) Programa basado en proceso



b) GUI basada en eventos

# Señales

## pyqtSignal

- `señal.connect(func)`  
Asocia la señal con una función.
- `señal.emit()`  
Gatilla el evento, es decir, envía la señal.
- `self.sender()`  
Permite acceder al elemento que gatilló el evento.

# Señales

 = `pyqtSignal()`

**A()**  
(emisor)

-----  
mensaje

**B()**  
(receptor)

 `.emit(mensaje)`

 `.connect(manejador)`

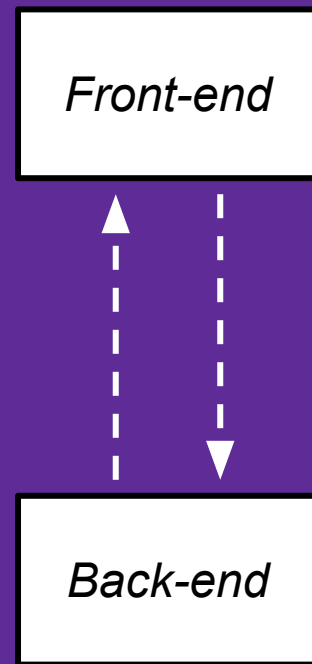
# Eventos predefinidos de PyQt6

- *MouseEvent*
- *KeyPressEvent*

---

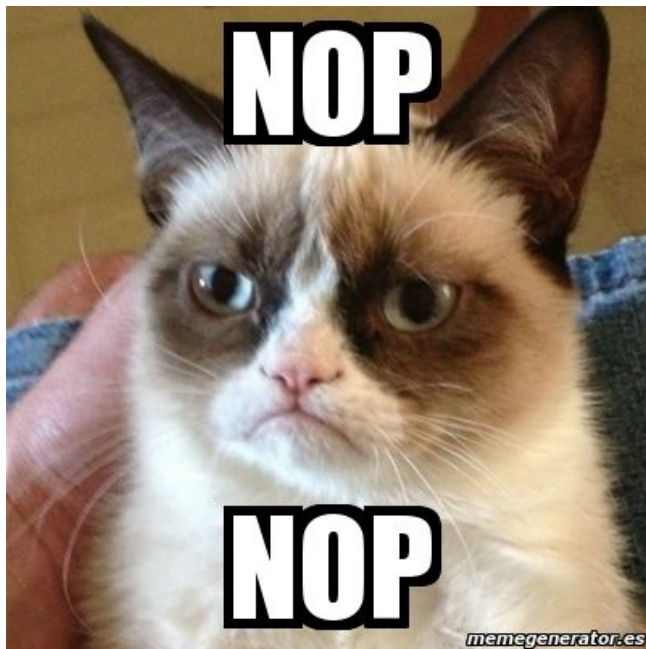


Tengan mucho  
cuidado con las  
dependencias  
circulares



# Tengan mucho cuidado con las dependencias circulares

```
class Frontend:  
    def __init__(self):  
        self.backend = Backend(self)  
  
class Backend:  
    def __init__(self, frontend):  
        self.frontend = frontend
```



# Tengan mucho cuidado con las dependencias circulares

```
class Frontend:
```

```
    def __init__(self):
```

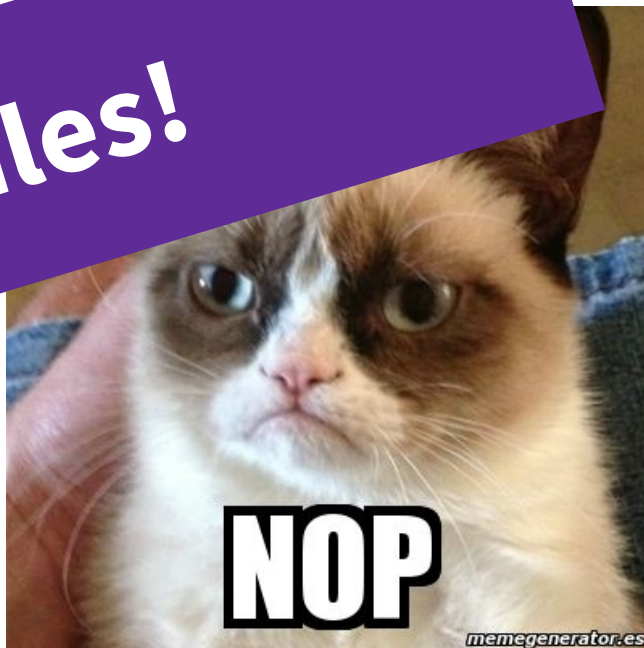
```
        self
```

```
class
```

**¡Usen señales!**

```
        self, frontend):
```

```
        self.frontend = frontend
```



# ***Programación Avanzada***

## **II C2233 2023-2**

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Joaquín Tagle - Francisca Cattán

