

Programación Avanzada

IIC2233 2024-2

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Lucas Van Sint Jan - Francisca Cattán



Experiencia 4

Interfaces Gráficas I

Experiencia 4: ¿Qué vamos a hacer?

1. Aplicaremos el patrón de diseño: *front-end/back-end*.
2. Crearemos una aplicación de múltiples componentes que interactúan entre ellos mediante el **uso de señales**.
3. Posicionaremos elemento en una ventana a través de:
 - a. **Coordenadas**
 - b. *Layouts*

DCChannels



A pocos días de Halloween 🎃🍬,
decides prepararte para tus días de
descanso y te propones crear tu
propio dispositivo para disfrutar de las
series y películas...

Un **televisor a control remoto.**

¿Cómo lo lograremos?

Programaremos una televisión, la cual estará compuesta por tres componentes:



Pantalla



Control
remoto



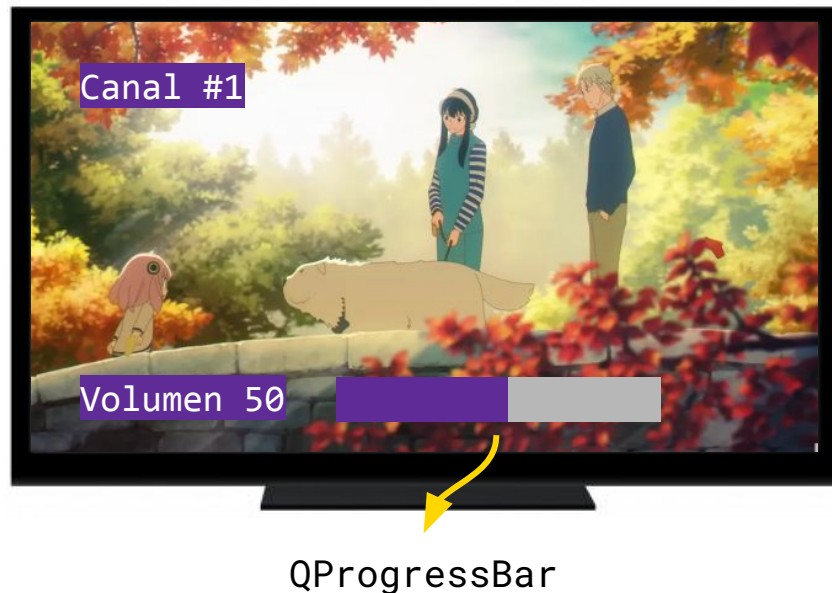
Controlador
lógico

Pantalla

Está compuesto por:

- 2 *Labels* para los textos
- 1 *Label* para la imagen de fondo
- 1 *ProgressBar* para mostrar el volumen.

Todos estos elementos serán posicionados de **forma manual** a través de **coordenadas**.



Control remoto

Está compuesto por:

- Múltiples botones (*PushButton*).
- 2 *Labels* para los textos de ciertos conjuntos de botones.

Todos estos elementos serán posicionados mediante ***layouts***.



Controlador lógico

- Se encargan de procesar toda la información lógica del programa.
 - Encender y apagar la tele.
 - Cambiar de canal.
 - Cambiar el volumen.
- Ayudan a mantener ciertos estados y su actualización.
- Permite la comunicación entre los distintos componentes visuales.

**Controlador
encargado
de la lógica**

¿Cómo lo lograremos?

Estos componente se encuentran incompletos o presentan errores, por lo que deberemos **completar o corregir los métodos** de cada componente:












- a. Pantalla
- b. Control remoto
- c. Controlador lógico

y del archivo **main.py**.
















¿Qué tenemos?

VentanaPantalla











Sin señales





-  posicion: tuple(int)
-  porte: tuple(int)
-  imagen: QLabel
-  canal: QLabel
-  volumen: QLabel
-  volumen_barra: QProgressBar
-  inicializar_gui()
-  generar_widgets()
-  agregar_estilo()
-  actualizar_volumen(volumen: int)
-  actualizar_canal(canal: int)

VentanaControlRemoto

-  senal_volumen: str
-  senal_canal: str
-  senal_encendido: null
-  volumen: list(QPushButton)
-  canales: list(QPushButton)
-  numeros: list(QPushButton)
-  inicializar_gui()
-  generar_botones()
-  generar_layout()
-  generar_layout_subir_bajar(
 botones: list, texto: str
)
-  generar_layout_numeros()
-  agregar_estilo()
-  conectar_botones()
-  actualizar_canal()
-  actualizar_volumen()

ControladorLogico

-  senal_volumen: int
-  senal_canal: int
-  senal_encendido: bool
-  senal_empezar: null
-  volumen: int (property)
-  canal: int (property)
-  prendido: bool
-  cambiar_volumen(cambio: str)
-  cambiar_canal(cambio: str)
-  actualizar_volumen()
-  actualizar_canal()
-  prender_apagar()
-  empezar()

 Señal
 Atributo
 Método
 Incompleto o no implementado

Parte 1: Completar componentes faltantes

Control remoto

Gran medida de sus *widgets* se encuentran definidos, pero **falta definir** el **botón ON/OFF** y los **botones de los números**.

Para esto, completa y corrige el método `generar_botones()`.

¡A programar! 



Parte 1: Completar componentes faltantes

Control remoto

Además, **falta posicionar los botones** para subir/bajar el canal y el volumen, **junto con el texto correspondiente**.

Deberás completar el método `generar_layout()`.



¡A programar! 

Experiencia 4

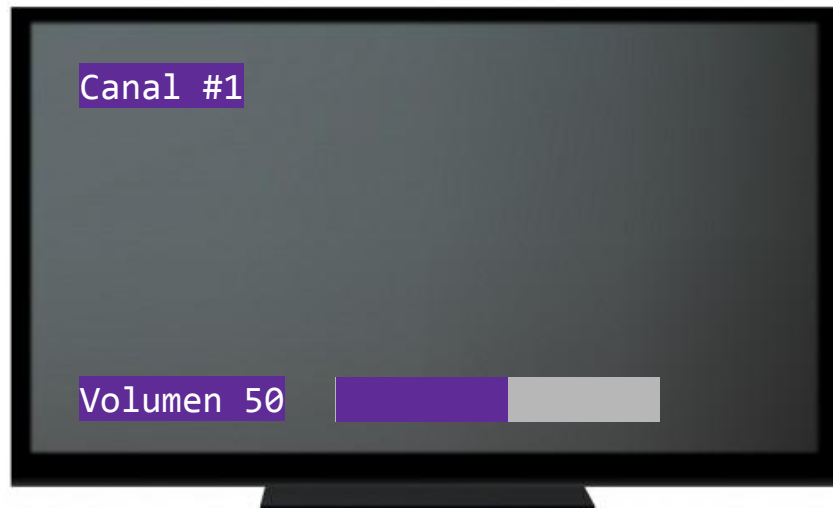
Esto es lo que tenemos hasta el momento:



Control
remoto



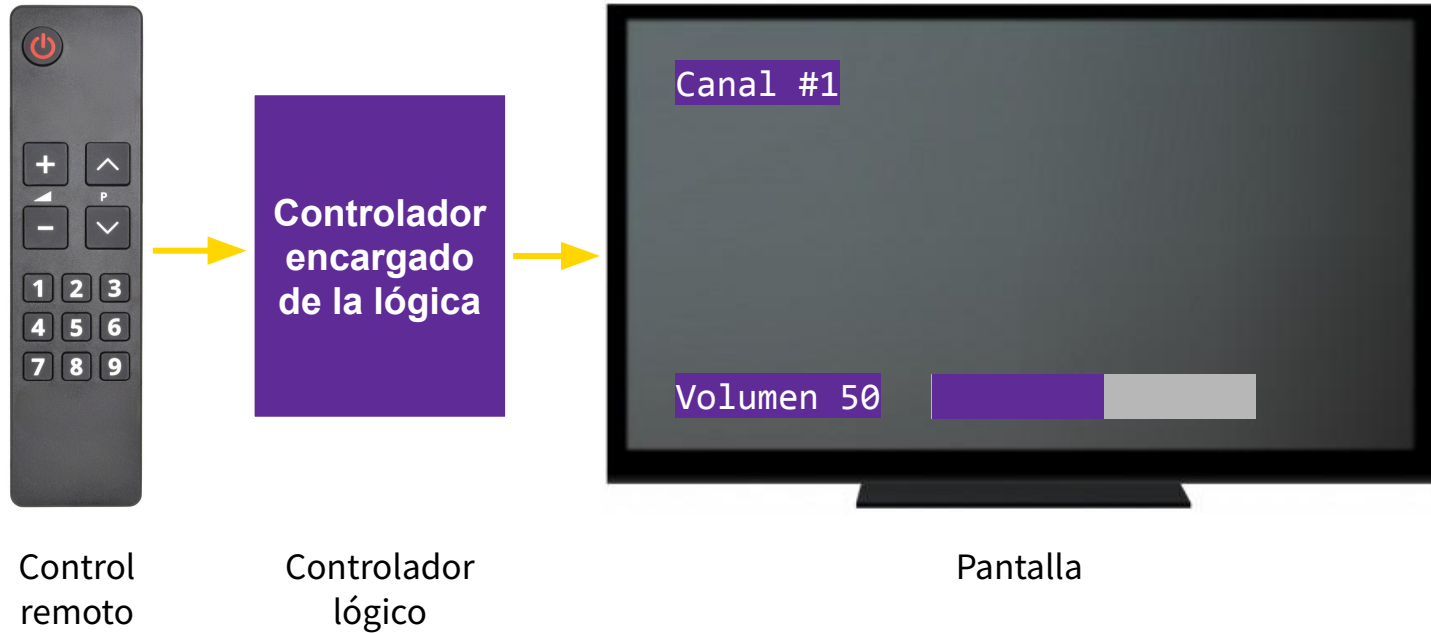
Controlador
lógico



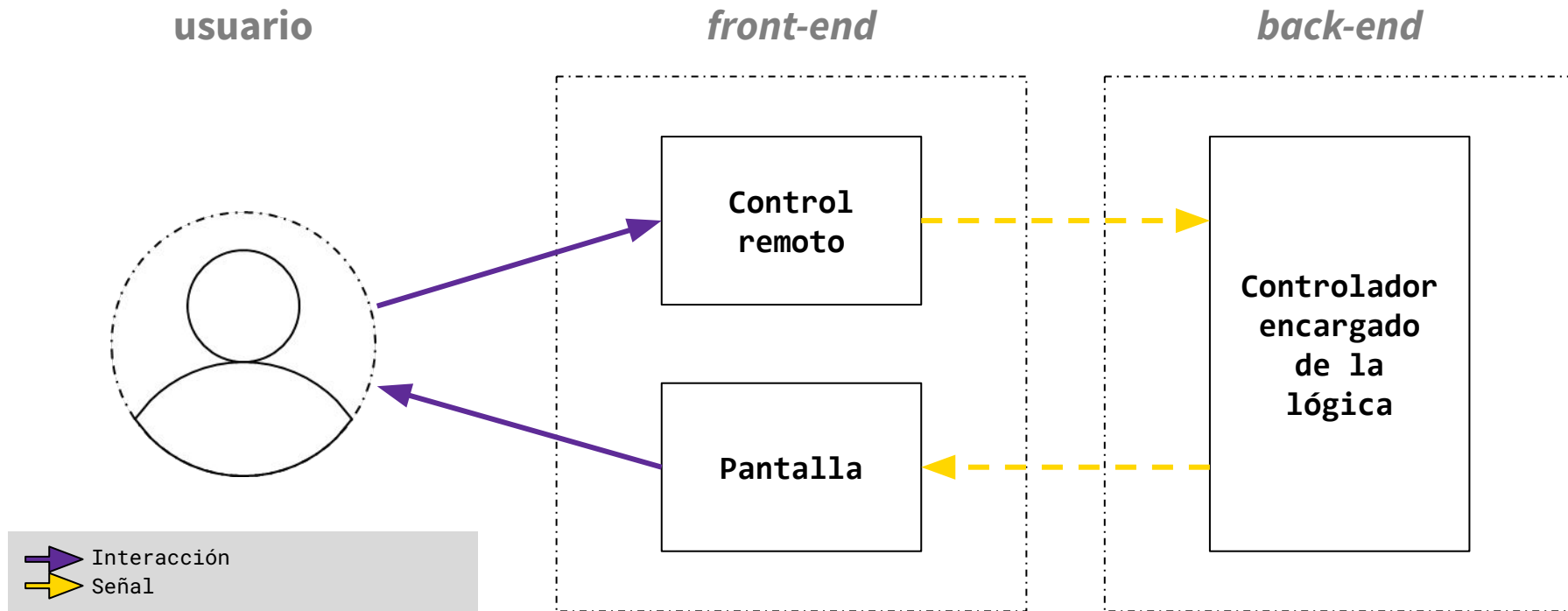
Pantalla

Experiencia 4

Falta permitir que los componentes se comunicarán entre ellos:



Experiencia 4: Diagrama de modelación



Parte 2: Enviar y manejar eventos

Los componentes visuales y lógicos ya se encuentran completos, solo falta:

1. **Enviar señales** para cambiar el volumen, canal y prender/apagar el televisor desde el **control remoto**:
 - a. `actualizar_canal()`
 - b. `actualizar_volumen()`

¡A programar! 

Parte 2: Enviar y manejar eventos

Los componentes visuales y lógicos ya se encuentran completos, solo falta:







2. **Recibir señales** para cambiar el volumen y canal del televisor en la **pantalla** y mostrar el cambio.
 - a. `actualizar_volumen(nuevo_volumen)`
 - b. `actualizar_canal(nuevo_canal)`

¡A programar! 










Parte 2: Enviar y manejar eventos

Los componentes visuales y lógicos ya se encuentran completos, pero presenta problemas. Ejecuta el archivo **main.py**, encuentra los errores y corrígelos.

VentanaControlRemoto

 senal_volumen: str	→	 cambiar_volumen(cambio: str)
 senal_canal: str	→	 cambiar_canal(cambio: str)
 senal_encendido: null	→	 prender_apagar()

ControladorLogico

 show()	←	 senal_volumen: int	→	 actualizar_volumen(volumen: int)
		 senal_canal: int	→	 actualizar_canal(canal: int)
		 senal_encendido: bool	→	 prender_apagar(encendido: bool)
		 senal_empezar: null	→	 show()

VentanaPantalla

¡A programar! 

Desafíos

Se pueden lograr con los **contenidos vistos hasta ahora**:

- Cambiar canal utilizando las teclas del teclado.
- Agregar un botón para “mutear”. Se debe recordar el volumen anterior.
- Agregar un ícono al botón “On/Off” y hacer que sea redondo.

Necesitamos de los contenidos que se verán en **Interfaces Gráficas 2**:

- Hacer *zapping*, apretando solo un botón.
- Agregar un protector de pantalla animado que se active después de cierto tiempo.
- Recibir canales de 2 dígitos o más, a través de los botones numéricos.

Programación Avanzada

IIC2233 2024-2

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Lucas Van Sint Jan - Francisca Cattán

