

[Año]

# [Título del documento]

[SUBTÍTULO DEL DOCUMENTO]

ALUMNO



# Índice

Ejercicio1:.....	1
•Explicación de cómo se ha organizado la aplicación.....	1
•Captura de funcionamiento.....	2
Ejercicio 2:.....	5
•Cómo funciona el cambio de informe mediante el ComboBox.....	5
•Qué criterio se ha seguido para asociar cada botón y cada elemento del ComboBox con su informe HTML.....	6
•Captura de funcionamiento.....	6
•Dificultades encontradas durante el desarrollo .....	9
Webgrafía.....	9

## Ejercicio1:

### •Explicación de cómo se ha organizado la aplicación.

La aplicación se ha estructurado utilizando la librería **PyQt5**.

**Diseño de la Interfaz (UI)**: He creado una ventana principal que hereda de QWidget. Para organizar los elementos visuales, he utilizado **QVBoxLayout**, lo que me permite apilar los botones de forma ordenada uno debajo del otro.

**Creación de Componentes**: He definido tres objetos **QPushButton** independientes. A cada uno le he asignado un texto descriptivo que identifica claramente el informe HTML que abrirá (DI\_U05\_A02\_03, DI\_U05\_A02\_08 y DI\_U05\_A03\_11).

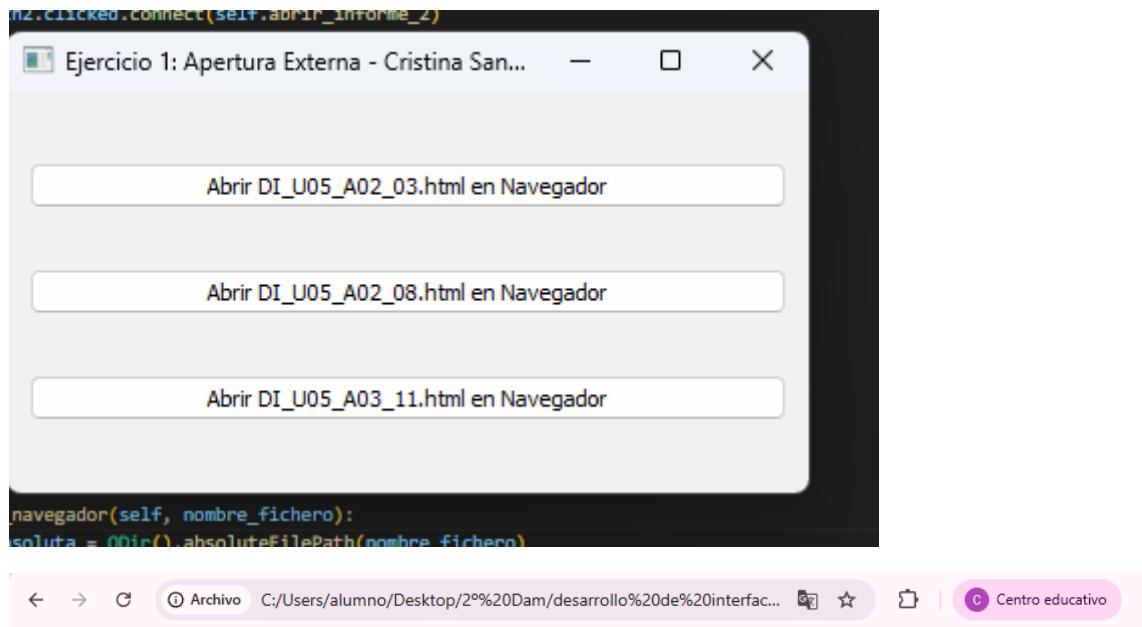
**Gestión de Eventos (Señales y Slots)**: He optado por un modelo de conexión directo para evitar funciones complejas. He conectado la señal clicked de cada botón a un método específico de mi clase.

**Lógica de Apertura Externa**: Para realizar la apertura en el navegador predeterminado, he implementado un método centralizado llamado `lanzar_navegador`. En este método he realizado dos pasos que considero fundamentales:

1. He obtenido la **ruta absoluta** del fichero usando `QDir().absoluteFilePath()`. Esto garantiza que la aplicación encuentre los informes sin importar desde qué carpeta se ejecute el script.
2. He utilizado **QDesktopServices.openUrl** junto con **QUrl.fromLocalFile**. He elegido esta vía porque es la forma estándar y más segura en Qt para invocar al navegador del sistema y mostrar archivos locales.

## •Captura de funcionamiento.

```
 sandoval_cristina_T4.5_E1.py > Ejercicio1 > lanzar_navegador
 1  #Cristina Sandoval Laborde 2ºDAM
 2
 3  import sys
 4  from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QPushButton
 5  from PyQt5.QtCore import QUrl, QDir
 6  from PyQt5.QtGui import QDesktopServices
 7
 8  class Ejercicio1(QWidget):
 9      def __init__(self):
10          super().__init__()
11          self.setWindowTitle("Ejercicio 1: Apertura Externa - Cristina Sandoval")
12          self.resize(400, 200)
13          self.initUI()
14
15      def initUI(self):
16          layout = QVBoxLayout()
17
18          # Creamos los botones uno a uno, que es lo más habitual
19          self.btn1 = QPushButton("Abrir DI_U05_A02_03.html en Navegador")
20          self.btn2 = QPushButton("Abrir DI_U05_A02_08.html en Navegador")
21          self.btn3 = QPushButton("Abrir DI_U05_A03_11.html en Navegador")
22
23          # Conectamos cada botón a una función que se encarga de su informe
24          self.btn1.clicked.connect(self.abrir_informe_1)
25          self.btn2.clicked.connect(self.abrir_informe_2)
26          self.btn3.clicked.connect(self.abrir_informe_3)
27
28          layout.addWidget(self.btn1)
29          layout.addWidget(self.btn2)
30          layout.addWidget(self.btn3)
31
32          self.setLayout(layout)
33
34          # Funciones específicas para cada informe
35      def abrir_informe_1(self):
36          self.lanzar_navegador("DI_U05_A02_03.html")
37
38      def abrir_informe_2(self):
39          self.lanzar_navegador("DI_U05_A02_08.html")
40
41      def abrir_informe_3(self):
42          self.lanzar_navegador("DI_U05_A03_11.html")
43
44      def lanzar_navegador(self, nombre_fichero):
45          ruta_absoluta = QDir().absoluteFilePath(nombre_fichero) |
46          QDesktopServices.openUrl(QUrl.fromLocalFile(ruta_absoluta))
47
48      if __name__ == "__main__":
49          app = QApplication(sys.argv)
50          ex = Ejercicio1()
51          ex.show()
52          sys.exit(app.exec_())
```



	Nombre	Mes	Unidades	Importe (€)
0	Renee Martin	Enero	242	4380
1	Angelica Armstrong	Enero	270	2213
2	Chadwick Mosley	Enero	325	5675
3	Boris Vazquez	Enero	343	4298
4	Halee Combs	Enero	431	2636
5	Kirsten Whitfield	Enero	581	5085
6	Halee Combs	Febrero	181	2800
7	Angelica Armstrong	Febrero	297	1144
8	Emerson Stephens	Febrero	490	1544
9	Renee Martin	Febrero	531	1025
10	Chadwick Mosley	Marzo	200	3086
11	Halee Combs	Marzo	253	3250
12	Kirsten Whitfield	Marzo	280	2519
13	Priscilla Hull	Marzo	384	5380
14	Angelica Armstrong	Marzo	559	1079
15	Renee Martin	Marzo	576	5776
16	Renee Martin	Abril	229	3257
17	Boris Vazquez	Abril	230	2734
18	Kirsten Whitfield	Abril	277	4909
19	Shad Hines	Abril	339	2128



## Informe de ventas

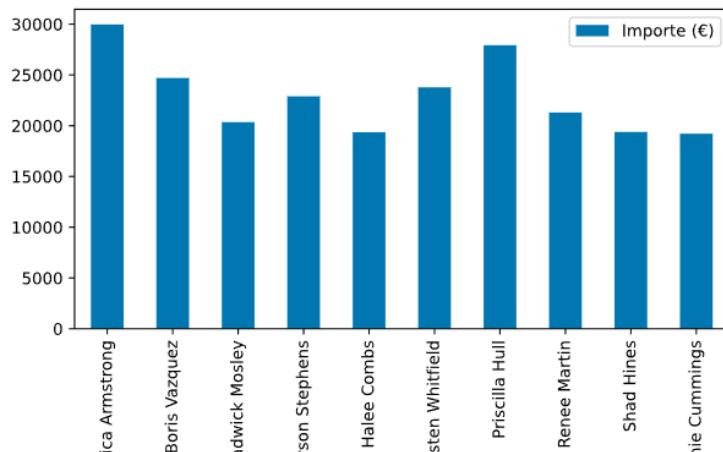
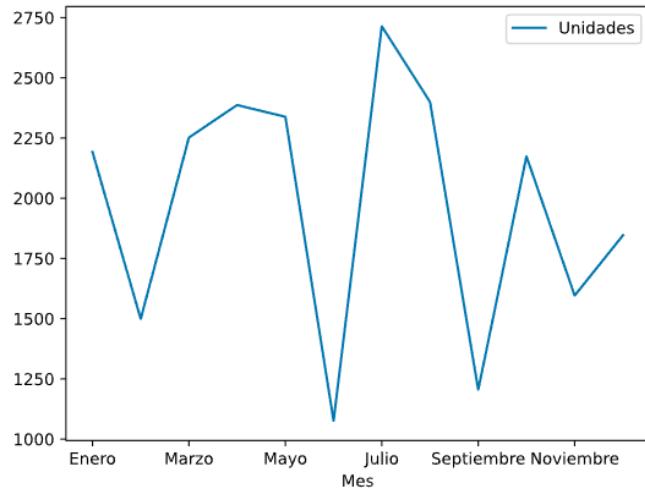
Unidades totales en diciembre

**1846**

Puedes descargar el fichero con los datos del informe.

ⓘ DI\_U05\_A02\_02.csv

Download



DI\_U05\_A03\_11 created on 5/3/2022, 13:56:12

datapane

## Ejercicio 2:

### •Cómo funciona el cambio de informe mediante el ComboBox.

**Poblamiento de datos:** He configurado el objeto QComboBox para que contenga los nombres de los tres archivos HTML requeridos como sus elementos de selección.

**Gestión de señales:** He conectado la señal currentTextChanged del ComboBox a mi método actualizar\_vista. Esto garantiza que, en el mismo instante en que selecciono un nombre distinto en el desplegable, la aplicación detecte el cambio.

**Carga del contenido:** El método actualizar\_vista recibe el nombre del fichero seleccionado, calcula su ruta absoluta mediante QDir y utiliza el método load() del componente QWebView para renderizar el HTML directamente dentro de mi ventana.

- Qué criterio se ha seguido para asociar cada botón y cada elemento del ComboBox con su informe HTML.

**Sincronización de nombres:** He definido los textos del ComboBox para que sean idénticos a los nombres reales de los archivos en el disco (DI\_U05\_A02\_03.html, etc.).

**Automatización:** Gracias a esta coincidencia, no he necesitado crear estructuras de datos intermedias como diccionarios o condicionales if/else complejos. Al seleccionar un elemento, el propio texto del ComboBox sirve como parámetro para localizar y cargar el archivo, haciendo que el código sea más limpio y eficiente.

- Captura de funcionamiento.

```
 sandoval_cristina_T4.5.E2.py > ...
1  #Cristina Sandoval LABORDE 2ºDAM
2
3  import sys
4  from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QComboBox
5  from PyQt5.QtWebEngineWidgets import QWebEngineView
6  from PyQt5.QtCore import QUrl, QDir
7
8  class Ejercicio2(QWidget):
9      def __init__(self):
10         super().__init__()
11         self.setWindowTitle("Ejercicio 2: Cristina Sandoval")
12         self.resize(1000, 750)
13         self.initUI()
14
15     def initUI(self):
16         layout = QVBoxLayout()
17
18         # Selector para cambiar el informe visualizado
19         self.combo = QComboBox()
20         self.informes = ["DI_U05_A02_03.html", "DI_U05_A02_08.html", "DI_U05_A03_11.html"]
21         self.combo.addItems(self.informes)
22         self.combo.currentTextChanged.connect(self.actualizar_vista)
23         layout.addWidget(self.combo)
24
25         # Componente para incrustar el contenido web
26         self.browser = QWebEngineView()
27         layout.addWidget(self.browser)
28
29         self.setLayout(layout)
30         # Carga inicial [cite: 51]
31         self.actualizar_vista(self.combo.currentText())
32
33     def actualizar_vista(self, nombre_fichero):
34         # Carga el informe seleccionado directamente en el componente
35         ruta = QDir().absoluteFilePath(nombre_fichero)
36         self.browser.load(QUrl.fromLocalFile(ruta))
37
38 if __name__ == "__main__":
39     app = QApplication(sys.argv)
40     ex = Ejercicio2()
41     ex.show()
42     sys.exit(app.exec_())
```

Ejercicio 2: Cristina Sandoval

DI\_U05\_A02\_03.html

	Nombre	Mes	Unidades	Importe (€)
0	Renee Martin	Enero	242	4380
1	Angelica Armstrong	Enero	270	2213
2	Chadwick Mosley	Enero	325	5675
3	Boris Vazquez	Enero	343	4298
4	Halee Combs	Enero	431	2636
5	Kirsten Whitfield	Enero	581	5085
6	Halee Combs	Febrero	181	2800
7	Angelica Armstrong	Febrero	297	1144
8	Emerson Stephens	Febrero	490	1544
9	Renee Martin	Febrero	531	1025
10	Chadwick Mosley	Marzo	200	3086
11	Halee Combs	Marzo	253	3250
12	Kirsten Whitfield	Marzo	280	2519
13	Priscilla Hull	Marzo	384	5380
14	Angelica Armstrong	Marzo	550	1070

Ejercicio 2: Cristina Sandoval

DI\_U05\_A02\_03.html  
DI\_U05\_A02\_03.html  
DI\_U05\_A02\_08.html  
DI\_U05\_A03\_11.html

	Nombre	Mes	Unidades	Importe (€)
0	Renee Martin	Enero	242	4380
1	Angelica Armstrong	Enero	270	2213
2	Chadwick Mosley	Enero	325	5675
3	Boris Vazquez	Enero	343	4298
4	Halee Combs	Enero	431	2636
5	Kirsten Whitfield	Enero	581	5085
6	Halee Combs	Febrero	181	2800
7	Angelica Armstrong	Febrero	297	1144
8	Emerson Stephens	Febrero	490	1544
9	Renee Martin	Febrero	531	1025
10	Chadwick Mosley	Marzo	200	3086
11	Halee Combs	Marzo	253	3250
12	Kirsten Whitfield	Marzo	280	2519
13	Priscilla Hull	Marzo	384	5380
14	Angelica Armstrong	Marzo	550	1070

Ejercicio 2: Cristina Sandoval

DI\_U05\_A02\_08.html



## Informe de ventas

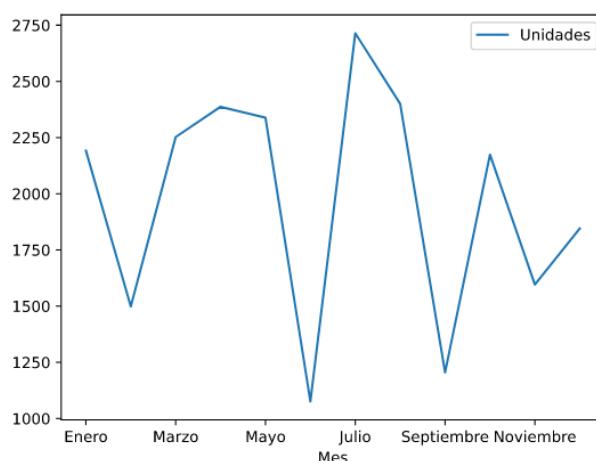
Unidades totales en diciembre  
**1846**

Puedes descargar el fichero con los datos del informe.

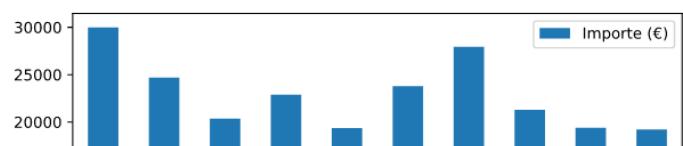
[@ DI\\_U05\\_A02\\_02.csv](#) [Download](#)

Ejercicio 2: Cristina Sandoval

DI\_U05\_A03\_11.html



A line chart titled 'Unidades' showing monthly sales units. The Y-axis ranges from 1000 to 2750 in increments of 250. The X-axis shows months from Enero to Noviembre. The data points are approximately: Enero (2400), Febrero (1500), Marzo (2300), Abril (2400), Mayo (2300), Junio (1100), Julio (2700), Agosto (2400), Septiembre (1250), Octubre (2200), Noviembre (1600), Diciembre (1846).



A bar chart titled 'Importe (€)' showing monthly sales import in euros. The Y-axis ranges from 20000 to 30000 in increments of 5000. The X-axis shows months from Enero to Noviembre. The data points are approximately: Enero (30000), Febrero (24500), Marzo (20500), Abril (22000), Mayo (23000), Junio (19500), Julio (24000), Agosto (28000), Septiembre (21000), Octubre (19500), Noviembre (19500), Diciembre (19500).

DI\_U05\_A03\_11 created on 3/5/2022, 1:56:12 PM

 datapane

- Dificultades encontradas durante el desarrollo**

**Rutas de archivo:** Inicialmente, los informes no se visualizaban porque el sistema buscaba rutas relativas. He tenido que asegurar el uso de QDir().absoluteFilePath() para que la aplicación funcione correctamente independientemente del directorio desde el que se lance.

## Webgrafía

<https://experts-deny-b9a.craft.me/5EVzmnvFZAgHAP>

<https://docs.datapane.com/quickstart/>

<https://doc.qt.io/archives/qtforpython-5/PySide2/QtWebEngineWidgets/QWebEngineView.html>