

Actividad - Calendario

A continuación, se trabajará en un widget muy común en muchas aplicaciones, el **calendario o calendar widget**. Observemos las siguientes imágenes:

Qt Proyecto Uno - ventana.ui*

Archivo Type Here

GESTIÓN CLIENTES

DNI: Fecha Alta:

Apellidos: Nombre:

Dirección: Provincia:

Sexo: ☐ Femenino ☐ Masculino Métodos de Pago: ☐ Efectivo ☐ Tarjeta ☐ Transferencia

Aceptar Salir

Object Inspector

Object

- MainWindow
 - centralwidget
 - btnAceptar
 - btnSalir
 - cmbProv
 - editApel
 - editData
 - editDir

Property Editor

Filter

MainWindow : QMainWindow

Property

- sizePolicy
- minimumSize
- maximumSize
 - Width
 - Height
- sizeIncrement
- baseSize

Qt Dialog - vncalendar.ui

abril 2020

	lu.	ma.	mi.	ju.	vi.	sá.	do.
14	30	31	1	2	3	4	5
15	6	7	8	9	10	11	12
16	13	14	15	16	17	18	19
17	20	21	22	23	24	25	26
18	27	28	29	30	1	2	3
19	4	5	6	7	8	9	10

En la primera imagen se observa una nueva **caja de texto y un botón** que en vez de texto tiene la imagen de un calendario. Para ello descargamos un icono de calendario de Internet y en la propiedad del botón **icon -> choose file** cargamos en el botón dicha imagen. Esa imagen se guardará en la carpeta **img** del proyecto.

Este botón tiene por función lanzar la ventana **venCalendar** encargada de contener el **widget Calendar**. Esta ventana es del tipo **dialog without buttons**. En este caso no es necesario preguntar al usuario si quiere cerrar la ventana ya que no es un evento muy necesario. En cuanto a su función, ésta es seleccionar una fecha, la de alta del cliente en este caso, y cargarla en el cuadro de texto que hay a su izquierda.

Estableceremos el evento con el código asociado.

Se declara en primer lugar la clase **DialogCalendar** en el **main.py** que nos permite lanzar la ventana del calendario. Además, empezamos a trabajar con la librería **datetime** de Python. El método **setselectedDate** nos permite seleccionar una fecha y el evento **cargaFecha** tiene como función cargar la fecha en la caja de texto de alta del cliente.

```
class DialogCalendar(QtWidgets.QDialog):
    def __init__(self):
        super(DialogCalendar, self).__init__()
        var.dlgcalendar = Ui_dlgCalendar()
        var.dlgcalendar.setupUi(self)
        diaactual = datetime.now().day
        mesactual = datetime.now().month
        anoactual = datetime.now().year
        var.dlgcalendar.Calendar.setSelectedDate((QtCore.QDate(anoactual,mesactual,diaactual)))
        var.dlgcalendar.Calendar.clicked.connect(clients.Cientes.cargarFecha)
```

En la clase **main.py** no olvidarse de **instanciar la clase anterior** y declarar el evento **abrirCalendar** que lanzará la ventana del calendario:

Instanciamos ambas clases:

```
class Main(QtWidgets.QMainWindow):
    def __init__(self):
        super(Main, self).__init__()
        var.ui = Ui_venPrincipal()
        var.ui.setupUi(self)
        var.avisosalir = DialogSalir()
        var.dlgcalendar = DialogCalendar()
```

Conectar el Botón con la función ***abrirCalendar***

```
'''
conexion de eventos con los objetos
estamos conectando el código con la interfaz gráfico
'''

QtWidgets.QAction(self).triggered.connect(self.close)
var.ui.btnSalir.clicked.connect(events.Eventos.Salir)
var.ui.actionSalir.triggered.connect(events.Eventos.Salir)
var.ui.editDni.editingFinished.connect(clients.Clientes.validoDni)
var.ui.btnCalendar.clicked.connect(clients.Clientes.abrirCalendar)
for i in var.rbtsex:
```

El módulo ***clients.py***, donde se sitúan los módulos ***abrirCalendar*** y ***cargaFecha***, será más o menos de la forma siguiente. Este módulo es ejecutado cuando se instancia `dlgCalendar`

```
'''
Abrir la ventana calendario
'''

def abrirCalendar():
    try:
        var.dlgcalendar.show()
    except Exception as error:
        print('Error: %s ' % str(error))

'''

Este módulo se ejecuta cuando clickeamos en un día del calendario, es decir, clicked.connect de calendar
'''

def cargarFecha(qDate):
    try:
        data = ('{0}/{1}/{2}'.format(qDate.day(), qDate.month(), qDate.year()))
        var.ui.editClialta.setText(str(data))
        var.dlgcalendar.hide()
    except Exception as error:
        print('Error cargar fecha: %s ' % str(error))
```