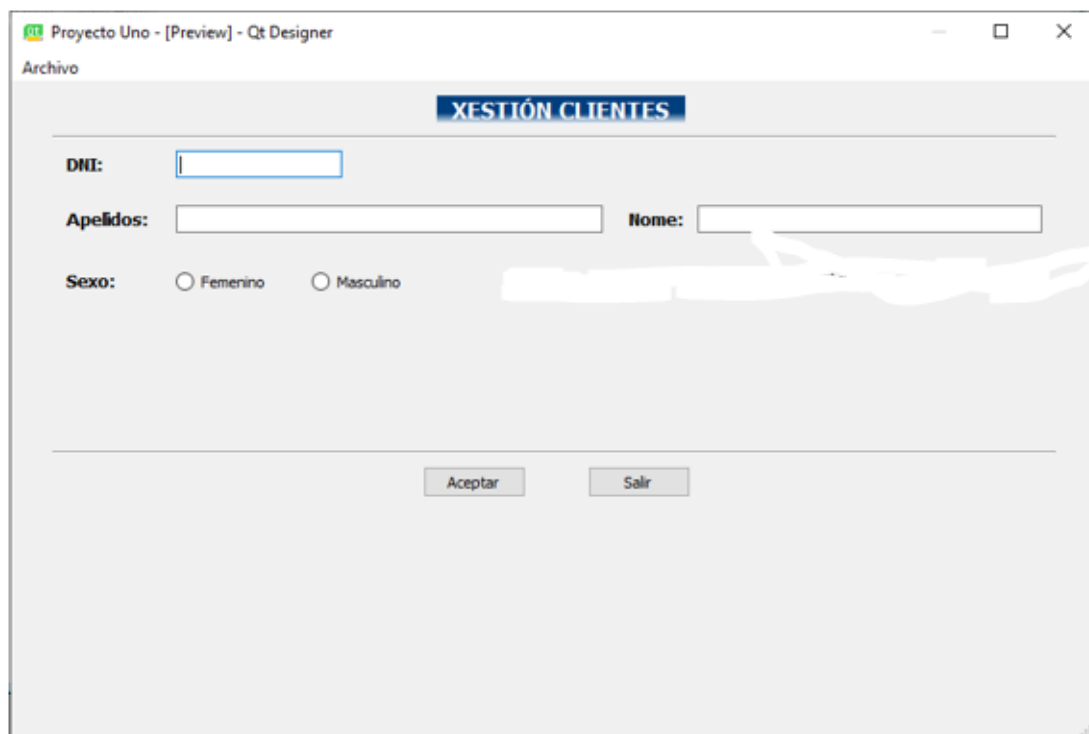


RadioButton

Los **RadioButton** son uno de los widgets más utilizados en los formularios, **son pequeños círculos**. Los utilizamos cuando queremos **seleccionar un solo elemento entre varios** elementos de un grupo. Su respuesta siempre es una variable booleana.

Veamos el siguiente ejemplo, donde un cliente puede ser varón o mujer, pero no ambos.



The screenshot shows a Qt Designer window titled 'Proyecto Uno - [Preview] - Qt Designer'. The form is titled 'XESTIÓN CLIENTES'. It contains the following elements:

- DNI:** A text input field.
- Apellidos:** A text input field.
- Nombre:** A text input field.
- Sexo:** Two radio buttons labeled 'Femenino' and 'Masculino'.
- Buttons:** 'Aceptar' and 'Salir' buttons at the bottom.

Algunas ideas sobre este diseño:

- utilizar layouts horizontales para agruparlos
- usad nombres intuitivos: *rbtFem*, *rbtHom*

A continuación, vemos las líneas de código que incluyen los eventos más importantes para el manejo de estos widgets.

```
conexión eventos botones
'''

#var.ui.btnAceptar.clicked.connect(events.Eventos.AltaCli)
var.ui.btnSalir.clicked.connect(events.Eventos.Salir)
var.ui.actionSalir.triggered.connect(events.Eventos.Salir)
var.rbtsex = (var.ui.rbtFem, var.ui.rbtMasc)
for i in var.rbtsex:
    i.toggled.connect(events.Eventos.selSexo)

'''

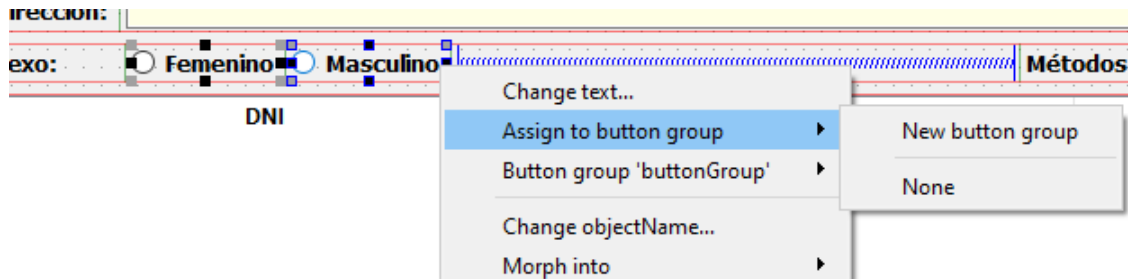
conexión eventos input widgets
'''
```

toggled es el evento que debemos llamar y ***isChecked*** es el método

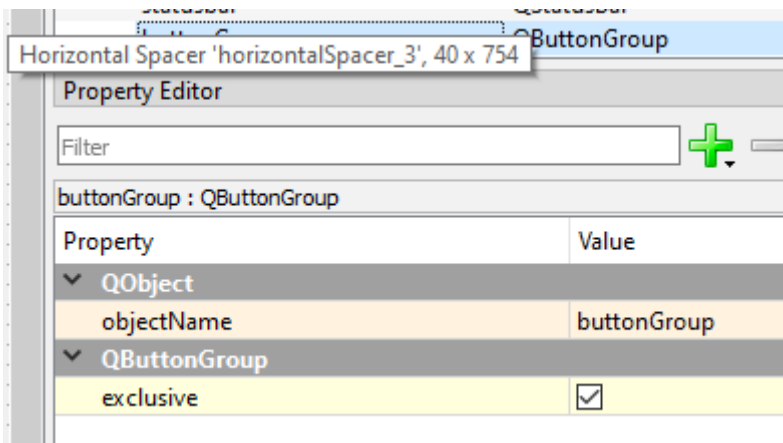
```
def selSexo(self):
    try:
        if var.ui.rbtFem.isChecked():
            print('marcado femenino')
        if var.ui.rbtHom.isChecked():
            print('marcado masculino')
    except Exception as error:
        print('Error en módulo seleccionar sexo:', error)
```

Para facilitar el manejo de los **radiobutton** se agrupan en los **QButtonGroup**.

Aunque es un widget, no aparece desde PyQt4 en la tabla. Para ello, nos situamos en el diseño marcamos los dos **radiobutton** y haciendo click con el botón derecho los añadimos a un **QButtonGroup**.



Se marcan los dos



Cambiamos el nombre y marcamos exclusive

A continuación, modificamos el código:

Eventos de botón

```
...  
var.ui.btnSalir.clicked.connect(events.Eventos.Salir)  
var.ui.rbtGroupSex.buttonClicked.connect(clients.Clientes.setSexo)  
...
```

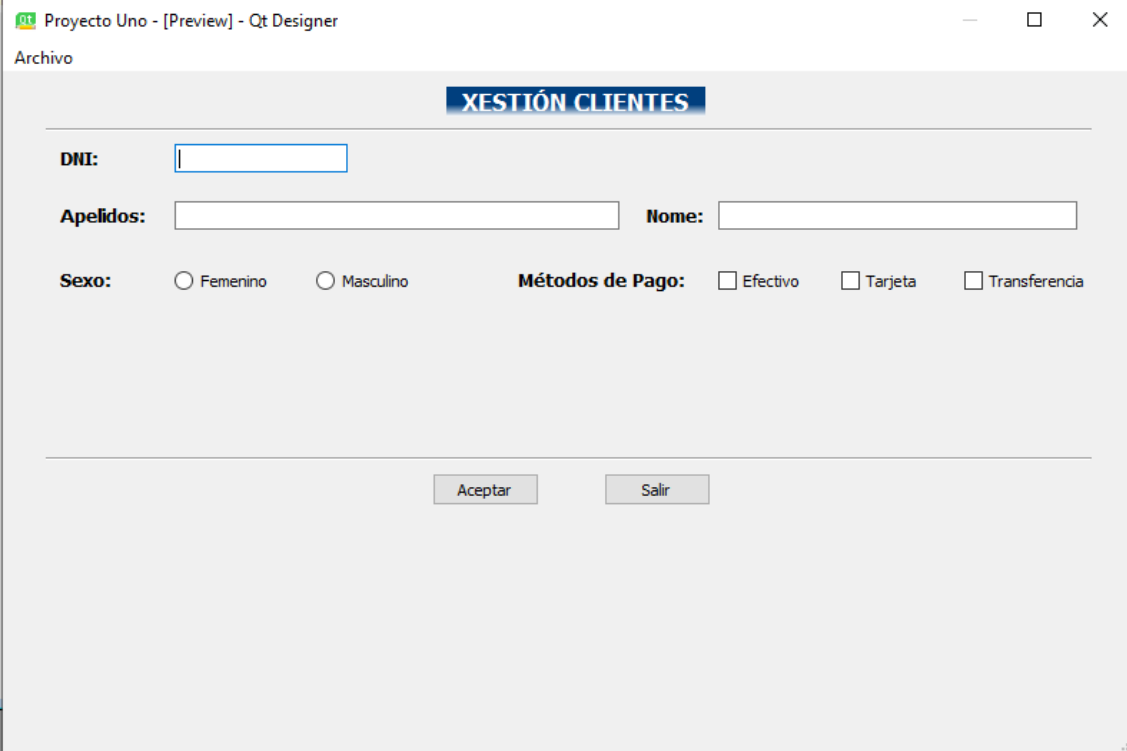
Checkbox

Junto con los **RadioButton**, los **Checkbox** son otro de los widgets más utilizados en los formularios.

A diferencia de los **RadioButton** (pequeños círculos), los **Checkbox** son **pequeños cuadrados**.

Los **RadioButton** los utilizamos cuando queremos seleccionar un solo elemento entre varios elementos de un grupo, los **Checkbox** se utilizan cuando necesitamos **tener uno o más elementos** seleccionados.

Continuando con el ejemplo anterior, un cliente puede utilizar diferentes tipos de métodos de pago según su deseo.



The screenshot shows a Qt Designer window titled 'Proyecto Uno - [Preview] - Qt Designer'. The form is titled 'XESTIÓN CLIENTES' and contains the following fields and controls:

- DNI:** A text input field.
- Apellidos:** A text input field.
- Nome:** A text input field.
- Sexo:** Two radio buttons labeled 'Femenino' and 'Masculino'.
- Métodos de Pago:** Three checkboxes labeled 'Efectivo', 'Tarjeta', and 'Transferencia'.
- Buttons:** Two buttons at the bottom labeled 'Aceptar' and 'Salir'.

Algunas ideas sobre este diseño:

- utilizar layouts horizontales para agruparlos
- usar nombres intuitivos: *chkEfectivo*, *chkTarjeta*, *chkTransf*

A continuación, vemos las líneas de código que incluyen los eventos más importantes para el manejo de estos widgets.

En el caso de los **checkbox**,

```
for i in var.rbtsex:
    i.toggled.connect(events.Eventos.selSexo)
var.chkpago = (var.ui.chkEfectivo, var.ui.chkTarjeta, var.ui.chkTransf)
for i in var.chkpago:
    i.stateChanged.connect(events.Eventos.selPago)
'''
```

stateChanged es el evento

```
def selPago():
    try:
        if var.ui.chkEfectivo.isChecked():
            print('pagas con efectivo')
        if var.ui.chkTarjeta.isChecked():
            print('pagas con tarjeta')
        if var.ui.chkTransf.isChecked():
            print('pagas con transferencia')
    except Exception as error:
        print('Error: %s ' % str(error))
```

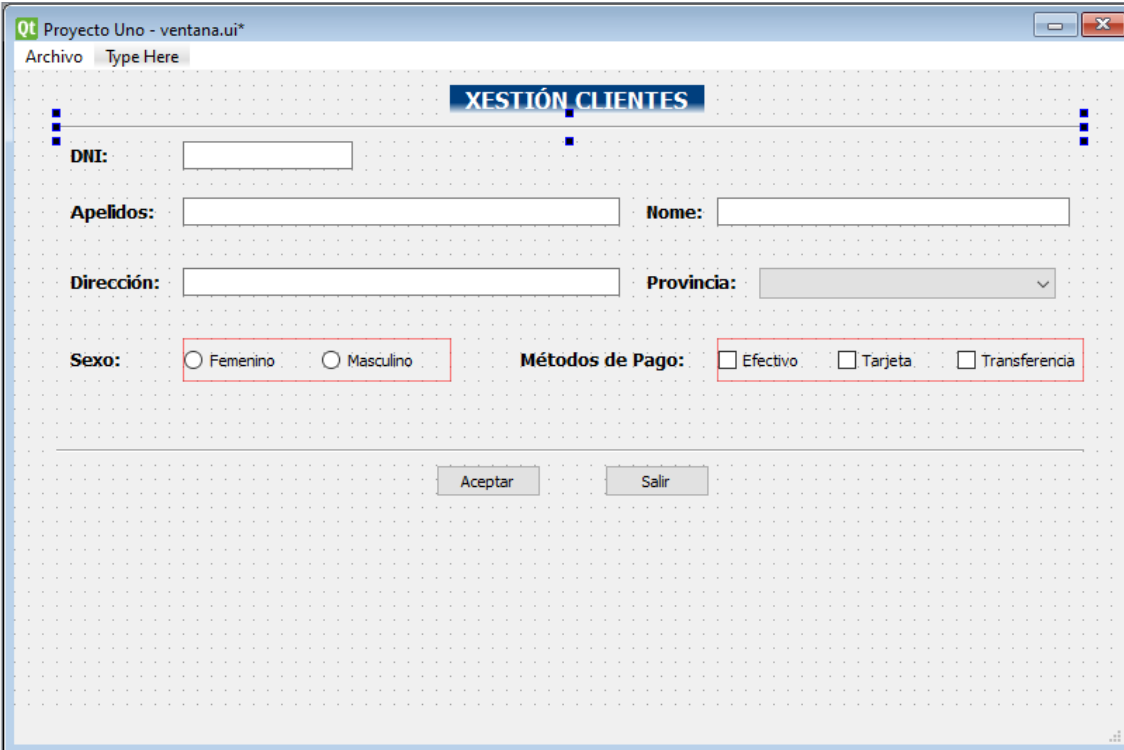
isChecked es el método

Combobox

El tercer widget más utilizado es el **combobox** o **cuadro combinado**.

Tradicionalmente, es una **combinación entre una lista desplegable y un cuadro de texto**, que permite al usuario introducir cualquier valor directamente en el teclado o, alternativamente, seleccionar un valor de la lista. El término "lista desplegable" se utiliza a veces para significar "cuadro combinado", aunque no todos los lenguajes lo consideran de la misma forma.

Modificar la IU para obtener el siguiente resultado:








The screenshot shows a Qt application window titled "Proyecto Uno - ventana.ui*" with a menu bar containing "Archivo" and "Type Here". The main content area has a title bar "XESTIÓN CLIENTES" and a form with the following fields:

- DNI:** A text input field.
- Apellidos:** A text input field.
- Nombre:** A text input field.
- Dirección:** A text input field.
- Provincia:** A combobox (dropdown menu).
- Sexo:** Two radio buttons labeled "Femenino" and "Masculino".
- Métodos de Pago:** Three checkboxes labeled "Efectivo", "Tarjeta", and "Transferencia".

At the bottom of the form are two buttons: "Aceptar" and "Salir".

Observamos los cambios en la interfaz gráfica: la inclusión de una caja de texto para introducir la dirección (por simplificar, obviamos cajas de texto para número de vivienda, piso, bloque, etc.). Además, supondremos que será en Galicia con lo que incluiremos en el cuadro combinado de la derecha solamente las provincias gallegas.

Esquema de los widgets introducidos hasta el momento. Obsérvese el nombre intuitivo dado en cada caso.

Object	Class
▼ MainWindow	QMainWindow
▼  centralwidget	QWidget
btnAceptar	QPushButton
btnSalir	QPushButton
cmbProv	QComboBox
editApel	QLineEdit
editDir	QLineEdit
editDni	QLineEdit
editNome	QLineEdit
▼  horlaySex	 QHBoxLayout
rbtFem	QRadioButton
rbtMasc	QRadioButton
▼  horlayPago	 QHBoxLayout
chkEfectivo	QCheckBox
chkTarjeta	QCheckBox
chkTransf	QCheckBox
lblApel	QLabel
lblDir	QLabel
lblDni	QLabel
lblMetpago	QLabel
lblNome	QLabel
lblProv	QLabel
lblSexo	QLabel
lblTitCli	QLabel
lblValido	QLabel
lineInf	Line
lineSup	Line
▼ menubar	QMenuBar
▼ menuArchivo	QMenu
actionSalir	QAction
statusBar	QStatusBar

Ahora trabajemos con el código del **combobox**:

```
i.stateChanged.connect(clients.Clients.selPago)

...
conexión eventos input widgets y otras funciones
...

clients.Clients.cargarProv()
var.ui.cmbProv.activated[str].connect(clients.Clients.selProv)
var.ui.editDni.editingFinished.connect(events.Eventos.validoDNI)

if __name__ == '__main__':
```

Líneas de código añadidos en **main.py**

```
def cargarProv():
    """
    esta solución es provisional en su momento lo haremos de otra forma
    cargando los registros desde una base de datos
    """
    try:
        prov = ['A Coruña', 'Lugo', 'Ourense', 'Pontevedra']
        for i in prov:
            var.ui.cmbProv.addItem(i)
    except Exception as error:
        print('Error: %s ' % str(error))
```

Función **cargarProv**

```
def selProv(prov):
    try:
        print('Has seleccionado la provincia de ', prov)
        return prov
    except Exception as error:
        print('Error: %s ' % str(error))
```

Función **selProv**

En el fichero **main.py** añadimos dos líneas:

- la primera ***clients.Clients.cargarProv()*** es una función que se lanza cuando arranca la aplicación. Su función es cargar el cuadro combinado o combobox.
- la segunda llama al evento **activated**. Este evento tiene como función cargar en una variable el valor seleccionado del combobox, de ahí que haya una tupla con un ***string***.

En el fichero **clients.py** tenemos dos funciones:

- la primera función ***cargarProv*** se encarga de cargar en el *combobox* el listado de provincias.
- la segunda función ***selProv*** nos devuelve la provincia seleccionada.